

**Do-it-yourself project to use a Wii remote controller, Wiiscan software and USB powering for a digital interactive whiteboard in classrooms**

---

# **Do-it-yourself project to use a Wii remote controller, Wiiscan software and USB powering for a digital interactive whiteboard in classrooms**

**Applies to Windows XP/Home, Vista and W7**

Author: Dirk Schouten, schoutdi (at) knoware (dot) nl, Public Primary School Rosa Boekdrukker, Amsterdam, the Netherlands, ict (at) rosaboekdrukker (dot) net, <http://rosaboekdrukker.net>.

## **Contents**

**Software update available: wiiscan-09-release-2.1.zip on <http://wyxs.net/downloads/whiteboard/>**

### 1. Introduction

1.1 Buying a set

1.2 Credits and acknowledgements

1.3 Warning!

1.4 Support

1.5 Skills

1.6 Conventions

### 2. What is needed in hard- and software

2.1 Our setup

2.2 USB 2.0 and Bluetooth

2.2.1 Computer

2.2.2 USB 2.0

2.2.3 Bluetooth

2.2.4 Hardware conclusions

2.3 Software

2.3.1 Service Packs

2.3.2 .NET Framework

2.3.3 Wiiscan

### 3. Building the USB power module & Wiimote powering

3.1 Schematics and printed circuit boards

3.2 Parts lists

3.3 Components

3.4 The building

3.5 Cables

3.6 Preliminary tests

3.7 Datasheets

### 4. Testing the USB power module

4.1 Determine the microcontroller version

[4.2 Installing the driver](#)

[4.3 Editing wiiscan.ini: power mode](#)

[4.4 Testing](#)

[4.5 Problems](#)

[5. Testing the Wiimote](#)

[5.2 Editing wiiscan.ini: MAC address](#)

[5.1 Connecting](#)

[5.3 Testing](#)

[5.4 Problems](#)

[6. Wiiscan](#)

[6.1 WIISCAN manual](#)

[6.2 WIISCAN -H.rtf](#)

[6.3 Wiiscan.ini](#)

[6.4 INSTALL.rtf](#)

[6.5 Wiiscantray](#)

[6.6 Basic pen software](#)

[6.7 Problems](#)

[7. Infra red pen](#)

[8. Whiteboards](#)

[8.1. Melamine](#)

[8.2. HPL \(High Pressure Laminate\)](#)

[8.2.1 Cleaning HPL](#)

[8.2.2 Glueing HPL](#)

[8.3. Enamel](#)

[8.4. Paint](#)

[8.5. Plastic foil](#)

[9. Speakers](#)

[9.1. Speaker in the projector](#)

[9.2. Separate speakers](#)

[10. Ceiling mounts](#)

[10.1. Wiimote ceiling mount](#)

[10.2. Video Projector ceiling mount](#)

[11. Video projector](#)

[12. Installing, connecting and testing](#)

[12.1 Whiteboard](#)

[12.2 Video projector](#)

[12.3 Wiimote](#)

[12.3.1 Distance of Wii to whiteboard](#)

[12.4 Speakers](#)

[12.5 Cables](#)

[12.6 Testing](#)

[12.7 Ready!](#)

[13. Software](#)

[13.1. IR pen](#)

[13.2. Pointers](#)

[13.3. Whiteboard software](#)

[14. Advanced tonics](#)

## [14. Advanced topics](#)

### [14.1 Preventing RF emission](#)

### [14.2 Working with the PixArt camera directly](#)

## [15. Testimonials](#)

## [16. Concluding remarks](#)

## [17. Disclaimer](#)

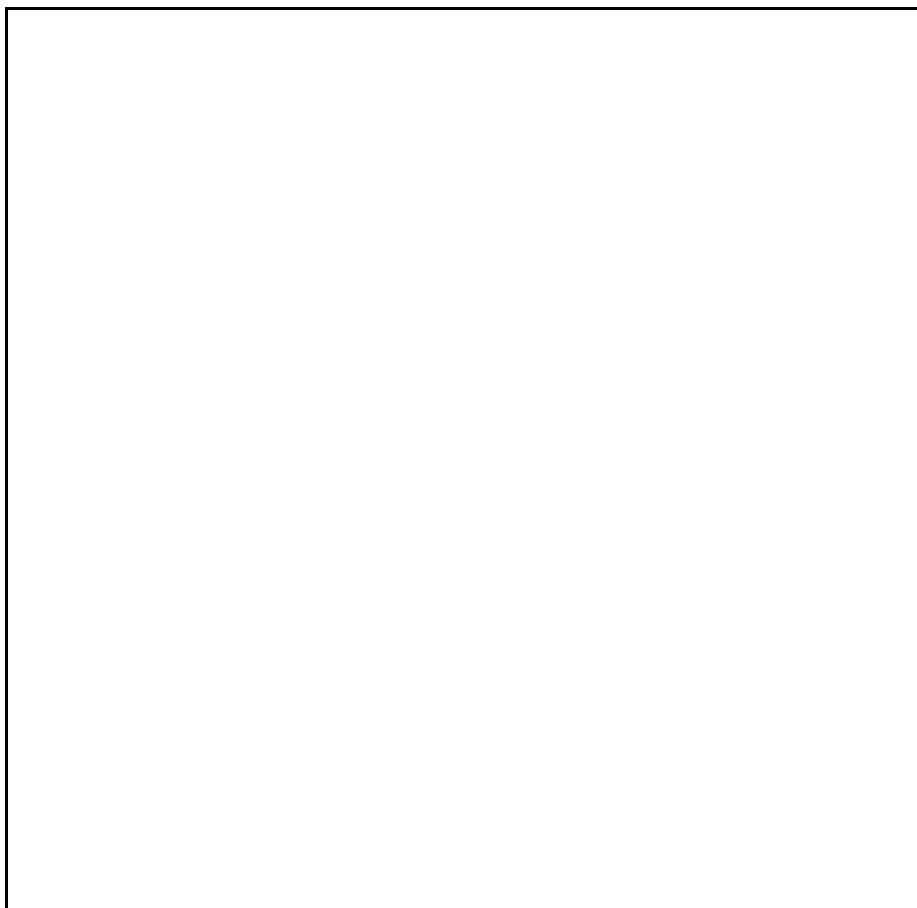
# 1. Introduction

The Nintendo Wii remote controller is a popular device to construct digital interactive whiteboards. Most of them are based on Johnny Chung Lee's original idea, to be viewed on <http://www.ted.com/talks/view/id/245>.

Johnny's splendid idea will almost certainly not function in a classroom, crowded with eager pupils and a teacher willing to educate them. Sooner or later one of them would touch the Wii remote controller (Wiimote from now on) , causing recalibration and thus the end of the lesson. Or, at the wrong moment of course, the batteries are empty and a pupil has to find fresh ones; unfortunately unavaiable. Or, starting up a program would fail because of some error, somehow during the operation. In classrooms Murpy's law is at its best! What is needed to use Johnnys idea in a classrooms is 'engineering'. The goal was set.

A small group of enthusiasts took up Johnnys initial idea to construct a digital interactive whiteboard that is effectively usable in a classroom. The how to of that first prototype is still available on [http://wyxs.net/web/wiimote/digital\\_whiteboard.html](http://wyxs.net/web/wiimote/digital_whiteboard.html). During a year a teacher of grade 3 and her pupils gained practical experience in the classroom with the prototype and discovered that it needed several improvements and engineering. The engineering involved designing a stable, robust, reproducable, error- and maintenance free, open solution. In the classroom it means that, after a complete installation, the teacher only needs to power on the PC to start wroking with the pen.

Below is a picture of the hardware:





complete\_set-460x613.jpg

## 1.1 Buying a set

When your school wants to buy complete sets for Windows (XP Pro/Home/ Vista or W7), either assembled and tested by the engineer Niels Kooy or as a do-it-yourself kit, please go to Niels's site:

<http://onsbord.nl/> (online and available October 2009)

where you can find ordering information and all other details.

A complete set consists of:

- Modified Nintendo Wii remote controller.
- Power clip to keep buttons pressed.
- Ceiling mount for the Wii remote controller.
- 10 meter cable with connectors, long enough to power the Wii hanging from the ceiling in a classroom to a PC in the classroom.
- USB power module, to power the Wii remote controller.
- USB cable (1 meter) to connect the USB power module to the PC.
- A professional infra red pen '[the Groove](#)'.
- The installation manual you are reading now.
- Installation instructions.

For those who already have some of the expensive parts, a do-it-yourself kit without them is available. It contains:

• USB power module, to power the Wii remote controller.

• USB cable (1 meter) to connect the USB power module to the PC.

• A professional infra red pen '[the Groove](#)'.

• The installation manual you are reading now.

• Installation instructions.

For those who already have some of the expensive parts, a do-it-yourself kit without them is available. It contains: the USB power module enclosure, connectors, printed circuit boards, a microcontroller, transistor, crystal, voltage regulator, resistors, capacitors, etcetera. Not included are the Nintendo Wiimote, the infra red pen, the ceiling mount and cables.



diy-kit.png

By supplying the do-it-yourself kit, we assume you have a Wiimote laying around and you can make an infra red pen yourself. Simple manuals can be found on many places on the internet and here: [http://wyxs.net/web/wiimote/digital\\_whiteboard.html](http://wyxs.net/web/wiimote/digital_whiteboard.html).

The cables we use are a standard USB A B cable between the Wiimote and the power module, and a self made cable with 2 RJ11 6P6C connectors and Cat 5 cable. For details, see section [3.5 Cables](#).

A third option is to read this how to and do everything yourself. The how to describes the construction of a USB power module to power a Wiimote via the USB port of a PC and a microcontroller, a P-Channel MOS FET and a voltage regulator. The microcontroller costs \$ 8,-- and is available via [Delcom](#) as part no. [902370](#). The other parts can be bought at most electronic parts stores.

Also the connection of the Wiimote via Bluetooth to the PC is described in detail, as well as an installation guide on .NET USB 2.0, .NET Framework, Bluetooth and the software. The software used to operate the Wiimote is 'Wiiscan', developed as open source software under a General Public License by Carsten Frigaard from [Mergeit](#). The how to also gives suggestions on setting up the videoprojector, on whiteboard materials, speakers, mounts, cursors, etcetera.

## 1.2 Credits and acknowledgements

A project like this leans on many shoulders. Basic research was done by Carsten Frigaard from [Mergeit](#). Carsten

A project like this leans on many shoulders. Basic research was done by Carsten Frigaard from [Mergeit](#). Carsten developed the Wiiscan software and the USB power module. We are very thankful for his OSS/GPL contribution to education.

Niels Kooy engineered Carstens work into robust, stable hardware. The public primary schools Corantijn and Joop Westerweel financed the initial experiments for this project. Peter Fokker, Arnold v.d. Burger and many others contributed with advice and practical help.

Primary education itself thanks you all for your contributions!

Of course, the author of this DIY project is the only one responsible for the text, its typos and errors. Tips and suggestions are welcome.

### 1.3 Warning!

Please, before doing anything, read section [13. Disclaimer](#), the usual disclaimer.

### 1.4 Support

When looking for support for the building, please find a electronics hobbyist. She is perfectly able to understand this text and build the devices for your school. As for installation support: please read this how to. It's all we know. Tips and suggestions are welcome.

### 1.5 Skills

When you have a ready made set, you only need to install software for which you need average computer skills. You should be able to open the DOS box and use basic DOS commands like 'cd', 'dir' and 'copy'. Furthermore you can download from the Internet, save, unzip, move and rename directories and files, install simple software, edit files with Notepad and create directories and links.

For the DIY project of making the USB power module and the conversions to the Wiimote, basic electronic skills are necessary. With the help of the soldering mask on the pc boards and the schematics, you will find your way. Other DIY skills needed are cutting, sawing, filing and soldering.

The most needed skills for this project are precision and patience, patience.

### 1.6 Conventions

A lot of work is done in Windows. To easily navigate in this OS we follow the way described in the following example, whereafter the explanation follows.

"Go Start > Settings > Control Panel > System > W: System Properties > T:Hardware > B: Device Manager > W: Device Manager > Universal Serial Bus Controllers > + > USB Root Hub.

This means: With your mouse, point to the 'Start' button in the lower left corner of your desktop. Left click with your mouse on the 'Start' button and navigate to 'Settings', whereafter you navigate to 'Control Panel'. Left click on 'Control Panel'. A new window opens: Select 'System'. A window (System Properties) opens, select tab 'Hardware'. The tab opens, click button 'Device manager'. Window Device manager Opens, select USB Controllers, click + to open the list and find 'USB Root Hub.

Using the '>', simple abbreviations and a bit of imagination, you will soon be familiar with this.

When we ask you to click a button, the button name is, for example, written as: [OK], [Next] or [Finish].

The text below is an adaption from one of our other projects: <http://serveratschool.net>

Part of the workstation setup is working on the command line in a DOS environment. Here are the rules of engagement:

- DOS is not case sensitive. When DOS commands, filenames and program names are used in the text, UPPERCASE haracters are used. This does not mean you have to *type* unpercase. Commands etc. can be issued

- DOS is not case sensitive. When DOS commands, filenames and program names are used in the text, `UPPERCASE` haracters are used. This does not mean you have to *type* uppercase. Commands etc. can be issued in lower or upper case, even mixed.
- Windows is not case sensitive but is case preserving. When Windows-specific names are used in the text, we try to preserve the case, e.g. `H:\My Documents` rather than the otherwise equivalent `H:\MY DOCUMENTS`.
- In DOS, commands are invoked with an `[Enter]`. We will omit "... and press the `[Enter]` key on your keyboard" with every instruction in order to keep the documentation as readable as possible.
- The names of buttons are displayed between square brackets, e.g. `[OK]`, `[Next]`, `[Finish]`.
- The keys on the keyboard are displayed as follows: `[Enter]` for the 'Enter' key and `[Tab]` for the tabulator key. Combinations of keys are displayed as follows: `[Alt-U]` means: 'press and hold the Alt key and hit the key labeled U' and `[Ctrl-Alt-Del]` means: 'press and hold the Ctrl key *and* the Alt key and hit the Del key'.

[\(top\)](#)

## 2. What is needed in hard- and software

This section discusses how to 'upgrade' your hard- and software to the level so that you can make use of the software for the digital interactive whiteboard, i.e. wiisan and pen software.

In this section we assume you already have a PC *with* Windows XP, *without* USB 2.0 and Bluetooth.

### 2.1 Our setup

This project was carried out on a Compaq D500 SFF with a Pentium IV 1,8 GHz processor, 512 Mb RAM and 4 *USB 1* slots. On the PC Windows XP Pro with Service Pack 3 was installed.

NOTICE:

The amount of memory (512 Mb RAM) is seems rather low. More is better; wich also applies to the processor speed. It seems a full fledged interactive whiteboard with running applications needs a rather fast computer. More info needed.

### 2.2 USB 2.0 and Bluetooth

#### 2.2.1 Computer

You need a PC with at least two free USB 2.0 slots. One for a USB Bluetooth radio (dongle) that establishes the data connection between the Wiimote and the PC. The other USB slot is used to power the USB power module that powers the Wiimote.

To check if the PC indeed has USB 2.0, proceed as follows:

Go: Start > Settings > Control Panel > System > W: System Properties > T: Hardware > B: Device Manager > W: Device Manager.

In the hardware list, search for 'Universal Serial Bus Controllers'. Click the + to expand the list. Look for the line containing 'XXX USB Enhanced Host controller'. 'XXX' refers to the chip brand.

More details on this subject are in [http://www.usbman.com/Guides/checking\\_for\\_usb\\_2.htm](http://www.usbman.com/Guides/checking_for_usb_2.htm), where is written: "If your Device Manager shows an XXX *Enhanced* USB Host Controller, the system has High Speed USB (USB 2.0) capability".

We did not found 'Enhanced' so we had to upgrade our hardware.

#### 2.2.2 USB 2.0

When the PC, like ours, has USB 1, you have to upgrade the PC to USB 2.0. We installed in the PC a 'Gembird' PCI controller card with 4 USB 2.0 slots. The card has a VIA chipset [\[1\]](#). It caused problems working in conjunction with our DHCP server. Later on we took Mr. Harsh advice serious and bought a Belkin USB 2.0 Hi-

conjunction with our DHCP server. Later on we took Mr. Harsh advice serious and bought a Belkin USB 2.0 Hi-Speed PCI card

Here is how to install and check a PCI controller card:

- Unplug the PC from mains. Remove the cover.
- Unscrew and remove the small metal cover from a free PCI slot.
- Carefully insert the USB controller card in the free PCI slot. Fix the card with the screw, close the cover, reconnect mains and power up.
- Several text balloons pass by: "Found new hardware". Let them pass by and wait until no more messages appear.
- It's Windows, so it's a good idea to restart the PC.

Check your work. Go: Start > Settings > Control Panel > System > W: System Properties > T: Hardware > B: Device Manager > W: Device Manager > Universal Serial Bus Controllers > +. Look for the line containing 'XXX USB Enhanced Host Controller'. 'XXX' refers to the chipset manufacturer.

In the list of USB controllers our 'VIA USB Enhanced Host Controller' was found. So, you now have USB 2.0.

### 2.1.1 Problems

Everything worked well with the PCI card, but we could not enter the BIOS anymore and we could not enter SABRINE (<http://serveratschool.net>). The problem was solved with the Belkin USB 2.0 fHigh-Speed PCI card. We also found found usefule info on [http://www.usbman.com/Guides/usb\\_2%20pci%20cards.htm](http://www.usbman.com/Guides/usb_2%20pci%20cards.htm) from which we quote: "VIA has entered the USB 2.0 PCI card business. VIA based cards can be troublesome and the driver installation is confusing. I recommend that you choose an NEC based card from the list of tested and approved cards below.". As said, we changed to Belkin. See the site.

Again, see [\[1\]](#)

In the next section we will add Blueooth functionality.

### 2.2.3 Bluetooth

The Wiiscan software steers the USB power moudle that powers the Wiimote. Wiiscan also takes care of setting up the connection between the Wiimote and the PC. The data transfer between the Wiimote and the PC is done via a Bluetooth connection. The connection is established with a USB Bluetooth device, also known as a 'dongle'. Basic info on Bluetooth can be found on: <http://en.wikipedia.org/wiki/Bluetooth>.

Info on the Bluetoot stack (driver software) can be found at: [http://en.wikipedia.org/wiki/Bluetooth\\_stack](http://en.wikipedia.org/wiki/Bluetooth_stack)

For this project, the USB Bluetooth device *must* be a Microsoft comatipble USB Bluetooth dongle. In other words, it must be able to operate with the MS Bluetooth stack (driver software).

NOTICE:

Do NOT install the driver that is packed with a USB bluetooth dongle! Wiiscan only operates with the Microsoft BT stack. Do not worry if you already installed BT drivers; this issue is discussed below.

Maybe it's best to buy [\[1\]](#) a USB Bluetooth dongle, of which the spcifications says it needs no driver on Windows XP Pro/Home, Vista or W7. That one will almost surely use the built in MS Bluetooth stack.

The USB BT dongle installation procedure is not, but can be complicated. Three conditions can occur:

1. Your PC does not have Bluetooth. You have to buy [\[1\]](#) a Bluetooth USB dongle. Buy a MS compatible one and continue with 'Installing a MS compatible USB Bluetooth dongle'.
2. Your PC has an MS compatible Bluetooth USB dongle, but you already installed a driver. Continue with 'I already insted a driver for my MS BT USB dongle'.
3. You have a BT dongle built in the PC or separate, but it's not MS compatible and a driver is installed. See below under 'BT USB dongle problems'

#### 1. Installing a MS compatible USB Bluetooth dongle



### 1. Installing a MS compatible USB Bluetooth dongle

Your PC has no USB BT dongle and you bought [\[1\]](#) a MS compatible BT USB dongle. Maybe the dongle did not even have a driver. That's fine and a indication that it might be Windows XP compatible. In case the dongle was supplied with driver softeware, do NOT install the driver. Proceed as follows:

1. Power on the PC and log in with administrative rights.
2. Choose a USB 2.0 port for the BT dongle. Choose this port carefully and, once chosen, do not change it anymore. An installed stack 'remembers' the port.
3. With a felt marker, mark the port 'BT'.
4. Insert the USB BT dongle in the 'BT' USB port.
5. Several text balloons with 'Found New Hardware' flash on the desktop. The last one saying that it is installend and can be used.
6. The BT USB dongle is installed.
7. Reboot, as usual.

We do a test to make sure the BT dongle is properly functioning.

#### USB Bluetooth dongle test

To check your work, go: Start > Settings > Control Panel > System > W: System Properties > T: Hardware > B: Device Manager > W: Device Manager > Bluetooth Radios > +. A 'Generic Bluetooth Radio' and 'Microsoft Bluetooth Enumerator' are installed.

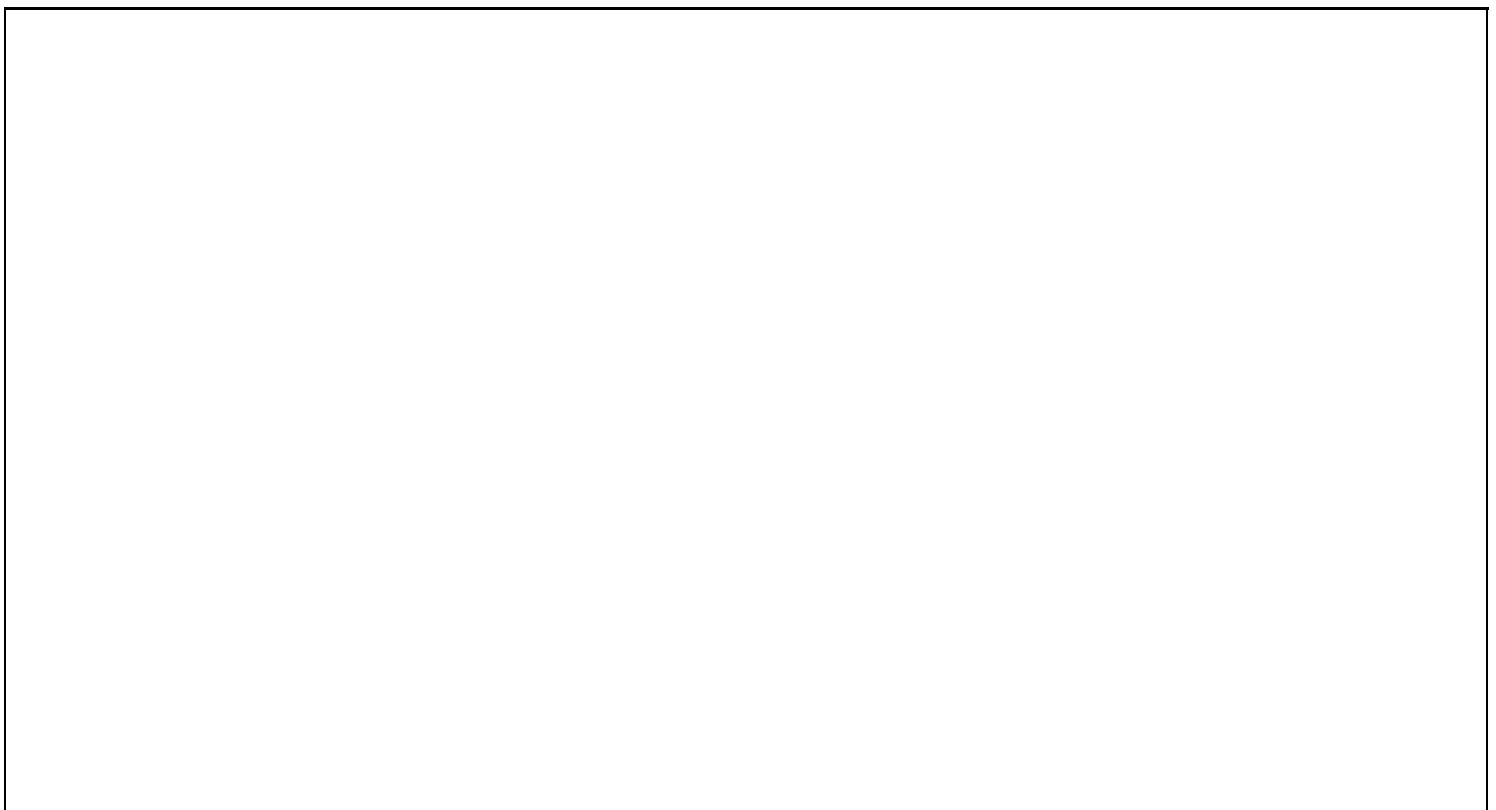
NOTICE: There are exceptions, we discovered. On an Acer Travelmate 5730 laptop, the generic BT radio was named "%Bluetoot USB BRCM Acer.device desc%". It worked file.

Also check if the dongle is actually working. You cannot use the USB power module because you are not sure it's built without errors. So you test Bluetooth functionality by using the Wiimote and power it with 2 1,5 Volt AA batteries in stead of the USB power module.

Carefully open the battery lid and remove it from the Wiimote. Insert two fresh 1,5 volt AA batteries in the battery compatiment.

Press the buttons 1 and 2 continuously; or even beetter, use a tie-wrap and a piece fo plastic to press the buttons 1 and 2 continuously. The four leds on the Wiimote will flash.

A permanent solution might be a piece of perspex as on this picture.





perspex\_wii\_permanent\_on.png

On your PC, go Start > Settings > Configuration Panel > Bluetooth > W: Bluetooth-devices > B: Add > W: Add Bluetooth Device Wizard > Check 'My device is set up and ready to be found' > B: Next > Swinging Torch icon ... wait... device is added: NINTENDO RVL-CNT 01.

Select the NINTENDO RVL-CNT 01; it becomes selected > Next > Check 'Don't use a passkey' > Next > W: Add Bluetooth Device Wizard > B: Finish.

Popup messages appear: Found New Hardware. The Wiimote has been found, proving the BT dongle is working. The 4 LEDss keep on flashing. In some cases only one LED flashes. Both are fine.

The window 'Bluetooth Devices' is still open. While you are here, you can do a necessary job of which the result is needed later on: obtaining the MAC (Media Access Control) address. It's a unique identification number that is given to a device in a network.

In the 'Bluetooth Devices' window, select the NINTENDO RVL-CNT 01 > B: Properties > W: NINTENDO RVL-CNT 01 Properties.

There you find the MAC address. It will be something like this example: Address: 00:21:bd:a4;07:ad. Your Wiimote's MAC address will differ from this example.

NOTICE: the letters are in lowercase here. Later on, you will use the MAC address with letters in UPPERCASE, thus the example above will read: 00:21:BD:A4;07:AD.

The MAC address cannot be cut and pasted, so write the MAC address on a piece of paper and double check if it's correct. Wiiscan will not work if the address does not match with the device.

## **2. I already installed a driver for my MS compatible USB BT dongle**

In this case you probably already had a USB dongle and self evidently you also install the driver. The dongle is MS compatible, and uses, as example, the installed Toshiba stack.

Proceed as follows:

compatible, and uses, as example, the installed Toshiba stack.

Proceed as follows:

1. Power on the PC and log in with administrative rights.
2. First check what you have. Go: Start > Settings > Control Panel > System > Hardware > Device Manager > List > Bluetooth Radio's > +. A lot of Toshiba entries in the list: Bluetooth RFNEP, RFBUS, RFCOMM, RFHID and the Generic Bluetooth radio.
3. A golden how to from <http://www.x-setup.net/forum/showthread.php?p=4762%3Cbr%3E> says: "Remove the stack by going: Start > Settings > Control Panel > Add/Remove Programs. Select the Toshiba driver and click [Remove]. Confirm that you are sure by clicking [OK]. The stack is removed".
4. Remove the stack.
5. Next, as usual, restart the PC and login with admin rights.
6. Insert the dongle. Several text balloons with 'Found new hardware' appear.
7. Go: Start > Settings > Control Panel > System > Hardware > Device Manager > List > Bluetooth Radio's > +. You see the 'Generic Bluetooth Radio' and the 'Microsoft-enumerator for Bluetooth'. Ok, case closed, the USB BT stack is properly installed.
8. In case you only find the 'Generic Bluetooth Radio', you right click on that line. A dropdown menu opens.
9. Select 'Update Driver'.
10. New window: update hardware. Check 'No, not now' and click [Next].
11. In the next window, select 'Install the software automatically (preferred)'. Click [Next].
12. In the next Window, click [Finish].

To check your work, perform the above test and click [here](#)

### 3. USB BT dongle problems

When you have a BT USB dongle with a stack, either on a PC with a PCI card, or built in a laptop, it might very well work with the Wiimote with batteries, but with the USB power module and Wiiscan or Wiiscantray it's not.

Help is given by Carsten Frigaard in his CHANGEINGBTSTACK.RTF document, found in C:\wiiscan\doc\. Below is a copy of the text. It's layout is slightly edited for browser viewing:

CHANGING the Widcomm BT stack to Microsoft BT stack

[From <http://forum.eeeuser.com/viewtopic.php?id=38533>]

After making my post in the 1000H post I thought it probably best if it here as I don't see why it wouldn't work on the 901 models either...

Basically it's all to just to change from using the Widcomm stack for using bluetooth connections to the built in Windows stack, it's fairly simple to do and should be a performance boost as Widcomm runs fairly sloppy.

basically if you follow a guide similar to:  
<http://www.x-setup.net/forum/showthread.php?p=4762>

Open file 'C:\Windows\inf\bth.inf' in notepad, (The inf folder is hidden by default so using Explorer go to Folder Options and change it to show all hidden folders)

(A good time to back up the bth.inf file if your worried about editing stuff wrongly but it's only like 3 lines worth)

Now you should have the section like this...

[Manufacturer]

```

CAMBRIDGE SILICON RADIO LTD.-CAMBRIDGE, NT.5.1
Dell=Dell, NT.5.1
FIC=FIC, NT.5.1
GVC=GVC, NT.5.1
HP=HP, NT.5.1
IBM=IBM, NT.5.1
Microsoft=Microsoft, NT.5.1
Motion Computing=MotionComputing, NT.5.1
Silicon Wave=SiliconWave, NT.5.1
Sony=Sony, NT.5.1
TDK=TDK, NT.5.1
TOSHIBA=Toshiba, NT.5.1
Wistron NeWeb=Wistron, NT.5.1
Zeevo=Zeevo, NT.5.1

```

Now add a line to the that section a line such as (without quotes of course):

```
'Azwave=Azwave, NT.5.1'
```

There should then be sections below this one that starts with:

```
;----- Device section - Start -----
```

Then insert a section that looks like this (Please note that you can check that the Vid and Pid are the same as your card by using device manage and looking at the properties of the device, and then the details tab, if not edit the following values, keeping the same format):

```
[Azwave.NT.5.1]
BT253= USB\Vid_0b05&Pid_b700
```

Then save and close the notepad.

Then just either do a 'I'll choose from a list of drivers install' or if you haven't installed the drivers yet do a disable and enable of the bluetooth and it should install it automatically. The driver won't be signed so continue with it anyways.

I've now tested it with receiving a file from a phone and as such it seems to work absolutely fine.

Updated due to my own incompetence, don't worry this fix still works great.

Last edited by Sly (2008-08-30 8:51:54 pm)

Changing the BT dongle seems the most simple solution.

Here are some more links, without guarantees. More tips on USB Bluetooth welcome.  
**NOTICE:** We experienced no problems with 2 'off the shelf' BT dongles.

- [http://www.tomshardware.co.uk/forum/page-128631\\_10\\_0.html](http://www.tomshardware.co.uk/forum/page-128631_10_0.html) Lots of info where we got several good links. And a quote: "FWIW, the Windows 98SE CD has a usbview.exe utility which identifies attached USB devices. And also: "And there is SIV, System Information Viewer that shows you directly what is on your USB buses, using a similar list". Google is your friend.

- <http://forum.eeeuser.com/viewtopic.php?pid=510064> Clear explanation of editing bth.inf.
- <http://www.usb.org/developers/tools/>. Scroll down to the "Company List. Download the Company List and save as usb.if. This is a long list with all VIDs (Vendor ID).
- <http://support.microsoft.com/kb/883259> Title: How to install and configure Bluetooth devices in Windows XP Service Pack 2. Comment: Basic information.
- <http://support.microsoft.com/default.aspx?scid=kb;en-us;840635> Title: Cannot install a Bluetooth device after you install Windows XP Service Pack 2. Comment: might be useful for editing BTH.inf.
- <http://support.microsoft.com/kb/841803>. Title: List of Bluetooth radio drivers that are included in Windows XP SP2. Comment: ...
- <http://forum.wiibrew.org/read.php?11,20081,20081> Cannot install a Bluetooth device after you install Windows XP Service Pack 2
- <http://joncellini.com/blog/archives/2004/10/07/further-adventures-with-bluetooth>. Here a different method is tried, copied from [http://www.technostan.com/archives/000068\\_windows\\_xp\\_sp2.html](http://www.technostan.com/archives/000068_windows_xp_sp2.html). That link does not exist anymore so here is an excerpt.

[quote] This got me to thinking if it would be possible to make my Ambicom dongle work with the Microsoft Bluetooth stack. After uninstalling the Widcomm drivers I was using and rebooting I found Windows would detect the Bluetooth adapter but complained that it did not have a suitable driver to support it. Fortunately with a bit of Googling, I was able to find a way to make it work simply by altering the contents of the Windows bth.inf file. The info here at Jon's Guides was pretty helpful to putting me on the right track. To make it work you simply need to add the applicable hardware IDs to the list of supported devices. Once this is done Windows will detect the adapter correctly and load the Microsoft Bluetooth stack. After the drivers are installed Bluetooth support is ready to use – in my case after bonding my Nokia 3650 phone to my PC (using the included wizard) I was able to fire up Nokia's PC Suite and everything worked automatically.

In case anyone is curious here is how I edited the bth.inf file to support my Ambicom adapter:

1. determine the hardware ID of your adapter. This can be done by clicking on the device in the Windows Device Manager and selecting "Hardware IDs" from the Details tab. My Ambicom device had a hardware ID of: vid\_0bdb&PID\_1002
2. [empty, DS]
3. once you have determined the hardware ID locate the bth.inf file and make a copy of it
4. open up the copy in an editor and scroll down to
5. the Device section
6. select one of the headings (I placed mine under [Belkin.NT.5.1]) and paste in the following:

```
; AmbiCom Support
Erickson Bluetooth Adapter= BthUsb, USBVID_0BDB&PID_1002
```

I named it as an Erickson Bluetooth Adapter since the chipset in my dongle appears to be made by Erickson.

7. Save the file
8. Update the dongle drivers to use your newly edited inf file. If all goes well it will detect and install the adapter.
9. Enjoy a working Bluetooth stack :)

[/quote]

- Our Belgian colleagues Kurt Dossche and David Braet (<http://digitaleschool.be>) mailed us:

When you have a laptop that has a built in BT dongle and an already installed driver (or a dongle with Xp-driver), this could be a workaround for problems:

- Go to Configuration Panel > System > Hardware > Bluetooth Devices > Bluetooth.
- At Properties, check Details and Device Instance ID
- You find something like: USB\VID\_0A5C&PID\_2101\5&2617BBE&02
- Write down this ID.

Now we will fool our laptop to use the MS Bluetooth stack and not the manufactureers stack:

- We install the standard XP driver which can be found in Windows directory windows/servicepackFiles/i386/.
- There you also see a file bth.inf.
- Open Notepad and load this file and search for:

```
;----- Device section - Start -----
```

- Scroll down to the point where you find something like:  
[Belkin.NT.5.1]  
Belkin Bluetooth Adapter= USBthUsb, USB\Vid\_050d&Pid\_0081  
Belkin Bluetooth Adapter= BthUsb, USB\Vid\_050d&Pid\_0084

- You add a line containing the previously noted Deice ID.
- Now you have someting like:

```
[Belkin.NT.5.1]
Belkin Bluetooth Adapter= BthUsb, USB\Vid_050d&Pid_0081
Belkin Bluetooth Adapter= BthUsb, USB\Vid_050d&Pid_0084
Belkin Bluetooth Adapter= BthUsb, USB\Vid_0A5C&Pid_2101
```

- Note that we do not write the backslash 'final'\5&2617BBE&02'.
- Save the bth.ini file on the same location.
- Finally, go to Configuration Panel > System > Hardware > BluetoothDevices > Bluetooth and remove the BlueTooth installed with the Manufactureer Driver
- Let Windows search for the new driver.
- If it is not found, point Windows to the right path, i.e. windows/servicepackFiles/i386/

- <http://www.shootingsoftware.com/Widcomm.htm> Title: Using the Microsoft Bluetooth Stack (instead of WIDCOMM) on Windows XP w/ SP2. Comment: good link.
- <http://forums.ni.com/ni/board/message?board.id=170&thread.id=405393>
- More links, relevance questionable:  
<http://www.x-setup.net/forum/showthread.php?p=4762>  
<http://forum.eeeuser.com/viewtopic.php?pid=336729>

## 2.2.4 Hardware conclusions

All hardware requirements seem rather easy to meet. The exception might be the USB Bluetoot dongle. The best way to avoid BT dongle problems is to make sure you have a dongle that is supported by the MS Bluetooth stack [1].

[1] The cocluding remark is from [Harsh J](#): "P.S. Next time, don't buy a cheap hardware [...]. Its better and easier to use a known and popular device even if they cost a few bucks extra. Or you can consider going in for Ubuntu OS rather".

We followed his advice and did not use a cheap USB 2.0 adapter card but bought a Belkin USB 2.0 Hi-Speed PCI card.

## 2.3 Software

For the USB power module and Wiiscan to work, you need Windows XP Pro/Home, Vista or W7 with at least Service Pack 2 and .NET Framework version 3.5 SP1. To check if the OS has these components, proceed as follows:

### 2.3.1 Service Packs

Go Start > Settings > Control Panel > System > W: System Properties > T: General.

You should see under 'System' a line: 'Service Pack 2'. Of course Service Pack 3 is also fine. Upgrade to at least SP2 and get all bugfixes and patches from MS. This can take a lot of time.

### 2.3.2 .NET Framework

Open Windows Explorer and type in the address bar: %systemroot%\Microsoft.NET\Framework

The directory might be empty, or you might see directories like: v1.1.4322, v1.0.3705, indicating the .NET Framework versions already installed on your OS.

Go to the MS site and download a version of .NET Framework 3.5 SP1 or abopve. Obey to MS rules. After downloading and installing, check if the version is properly installed by typing in the address bar in Explorer: %systemroot%\Microsoft.NET\Framework

Update your OS if necessary before proceeding any further with this project.

NOTICE: We upgraded to .NET Framework Version 3.5 SP1.

You also need the Wiiscan program and the pen software. These are discussed in the next section below and section [6.6 Basic pen software](#). Do not install that software now since a basic pen version is already in Wiiscan. The installation of the right version of the pen softwar is discussed in section [13.1 IRpen](#)

### 2.3.3 Wiiscan

- Download and install the Wiiscan software and documentation from Mergeit. Go to <http://mergeit.dk/open-it/projekter/egne/wii/> and download wiiscan-0.9-release-2.0.zip. This is the latest known version (July 2009). You can also download from <http://wyxs.net/downloads/whitebaord/>. Save the file, for example in C:\
- Unzip the wiiscan-0.9-release-2.0.zip file, for example in C:\. Rename the directory, omitting the quotes, in 'wiiscan' to reduce typing later on.

The unzipped and renamed directory contains the following directories and sub-directories:

```
C:\WIISCAN>tree /f
Map PATH-lijst
The volume number is 4A69-73E7
C:
+---redist
|       usbiods.inf
|       usbiods.sys
|
+---src
|   |   wiiscan-0.9.zip
|   |   wiimotelibpoll-0.9.zip
|   |   wiiscantray-0.9.zip
|   |
|   +---wiimotelibpoll-0.9
|   |   +---wiimotelibpoll-0.9
|   |   |       WiimoteWhiteboard.sln
|   |   |       wiimotelibpoll.csproj
|   |   |       Form1.cs
|   |   |       Form1.Designer.cs
|   |   |       wiimotelibpoll.cs
|   |   |
|   |   +---Properties
|   |   |       Settings.settings
|   |   |       Resources.Designer.cs
|   |   |       Settings.Designer.cs
|   |   |       AssemblyInfo.cs
|   |   |       Resources.resx
|   |
|   |
|   |
```

```
+---Debug
    wiiscan.ini
    wiiscan.exe
    WiimoteLib.dll
    DelcomDLL.dll
    wiimotelibpoll.exe
    USBm.dll
    wiiuse.dll

+---Release
    wiiscan.ini
    wiiscan.exe
    WiimoteLib.dll
    DelcomDLL.dll
    wiimotelibpoll.exe
    USBm.dll
    wiiuse.dll

+---src
    funs.h
    wiiscan.cpp
    wiiuse.h
    templatefun.h
    exception.h
    args.h
    process.h
    configfile.h
    reg.h
    usbm.h
    wiiscan.h
    stringfun.h
    DelcomDLL.h
    toolsfun.h
    file.h

+---Lib
    | DelcomDLL.lib
    | wiiuse.lib
    | DelcomDLL.dll
    | USBm.dll
    | wiiuse.dll
    |
    +---Debug
    |     WiimoteWhiteboard.
    |     WiimoteLib.dll
    |     wiimotelibpoll.exe
    |
    +---Release
    |     WiimoteWhiteboard.
    |     WiimoteLib.dll
    |     wiimotelibpoll.exe
    |

+---Doc
    MANUAL_wiiscan.rtf
    INSTALL.rtf
    CHANGEINGBTSTACK.rtf
    CHANGELOG.rtf
    LICENSE.rtf
    README.rtf
    wiiscan.1

+---Tools
    disable_all_usb.sh
    popup_disable.reg
    enable_all_usb.sh
```



```
+---wiiscantray-0.9
|
|   +---wiiscantray-0.9
|   |   wiiscantray.sln
|   |   wiiscantray.vcproj
|   |
|   |   +---Debug
|   |   |   WiimoteWhiteboard.exe
|   |   |   wiiscan.ini
|   |   |   WiimoteLib.dll
|   |   |   DelcomDLL.dll
|   |   |   wiimotelibpoll.exe
|   |   |   USBm.dll
|   |   |   wiiuse.dll
|   |   |   wiiscantray.exe
|   |   |
|   |   |   +---Release
|   |   |   |   WiimoteWhiteboard.exe
|   |   |   |   wiiscan.ini
|   |   |   |   wiiscan-release.ini
|   |   |   |   WiimoteLib.dll
|   |   |   |   DelcomDLL.dll
|   |   |   |   wiimotelibpoll.exe
|   |   |   |   USBm.dll
|   |   |   |   wiiuse.dll
|   |   |   |   wiiscantray.exe
|   |   |   |
|   |   |   |   +---src
|   |   |   |   |   systemtraysdk.h
|   |   |   |   |   resource.h
|   |   |   |   |   wiiscantray.cpp
|   |   |   |   |   systemtraysdk.cpp
|   |   |   |   |   wiiscantray.rc
|   |   |   |   |
|   |   |   |   |   +---Lib
|   |   |   |   |   |   DelcomDLL.lib
|   |   |   |   |   |   wiiuse.lib
|   |   |   |   |   |   DelcomDLL.dll
|   |   |   |   |   |   USBm.dll
|   |   |   |   |   |   wiiuse.dll
|   |   |   |   |   |
|   |   |   |   |   |   +---Debug
|   |   |   |   |   |   |   WiimoteWhiteboard.
|   |   |   |   |   |   |   WiimoteLib.dll
|   |   |   |   |   |   |   wiimotelibpoll.exe
|   |   |   |   |   |   |
|   |   |   |   |   |   |   +---Release
|   |   |   |   |   |   |   |   WiimoteWhiteboard.
|   |   |   |   |   |   |   |   WiimoteLib.dll
|   |   |   |   |   |   |   |   wiimotelibpoll.exe
|   |   |   |   |   |   |
|   |   |   |   |   |   |   +---Doc
|   |   |   |   |   |   |   |   INSTALL.rtf
|   |   |   |   |   |   |   |   CHANGEINGBTSTACK.rtf
|   |   |   |   |   |   |   |   CHANGELOG.rtf
|   |   |   |   |   |   |   |   LICENSE.rtf
|   |   |   |   |   |   |   |   README.rtf
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   +---res
|   |   |   |   |   |   |   |   |   wiiscan_06.ico
|   |   |   |   |   |   |   |   |   wiiscan_07.ico
|   |   |   |   |   |   |   |   |   wiiscan_01.ico
|   |   |   |   |   |   |   |   |   wiiscan_03.ico
|   |   |   |   |   |   |   |   |   wiiscan_05.ico
|   |   |   |   |   |   |   |   |   wiiscan_04.ico
|   |   |   |   |   |   |   |   |   wiiscan_00.ico
|   |   |   |   |   |   |   |   |   wiiscan_08.ico
```

```
|
+---doc
|   CHANGEINGBTSTACK.rtf
|   wiiscan.l
|   license.rtf
|   readme.rtf
|   MANUAL_wiiscan.rtf
|   CHANGELOG.rtf
|   install.rtf
|
+---bin
|   +---Release
|   |   WiimoteWhiteboard.exe
|   |   wiimotelibpoll.exe
|   |   WiimoteLib.dll
|   |   USBm.dll
|   |   wiiuse.dll
|   |   wiiscantray.exe
|   |   WiimoteWhiteboard_v0.3.exe
|   |   wiiscan.exe
|   |   wiiscan.ini
|   |   DelcomDLL.dll
|   |   calibration.dat
|   |   wiiscan_log.txt
|   |   xerror.txt
|   |
|   +---Debug
|   |   WiimoteWhiteboard.exe
|   |   wiimotelibpoll.exe
|   |   WiimoteLib.dll
|   |   USBm.dll
|   |   wiiuse.dll
|   |   wiiscantray.exe
|   |   WiimoteWhiteboard_v0.3.exe
|   |   wiiscan.exe
|   |   wiiscan.ini
|   |   DelcomDLL.dll
```

C:\WIISCAN>

NOTICE: The tree is boringly long, but in several directories you can find documentation! So it's a good idea to unzip all directories.

[\(top\)](#)

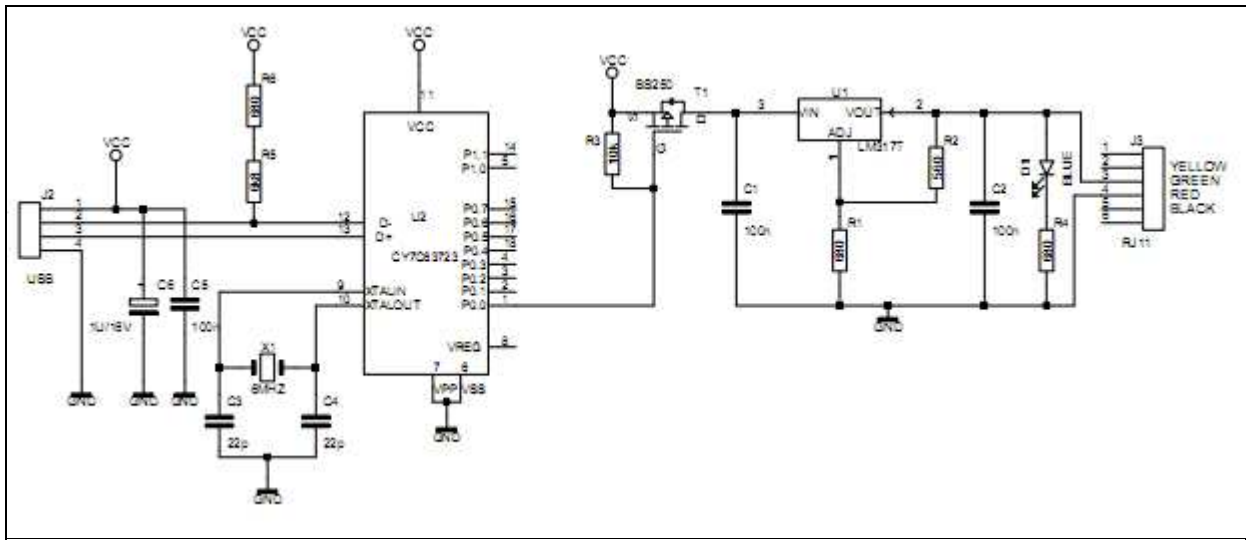
## 3. Building the USB power module & Wiimote powering

In this section we discuss the do-it-yourself version of the digital interactive whiteboard project. First we will discuss the electronic schematic and then the actual building and testing.

### 3.1 Schematics and printed circuit boards

Below are the schematics and PCB's to build the USB power module and the powering of the Wiimote.

- [Wiiscan\\_sch.pdf](#) is the schematic of the USB power module that powers the Wiimote via the PC's USB port. The schematic is based on basic research by Carsen Frigaard from [Mergeit](#). His work is adapted and engineered by Niels Kooy.



USB\_power\_module\_schematic.png

Discussion of schematic:

Via the USB data connection, the Delcom microcontroller (U2) gets data, and translates the commands from Wiiscan to an on/off signal. This signal is fed to the gate of the P-channel MOS FET transistor (T1) which is the real switch that turns the power to the Wiimote on/off.

In the schematic the gate of the P-channel MOS FET transistor (T1) is permanently pulled-up to 5 Volt by the resistor (R3), so the Wiimote gets no power and the LED (D1) is off.

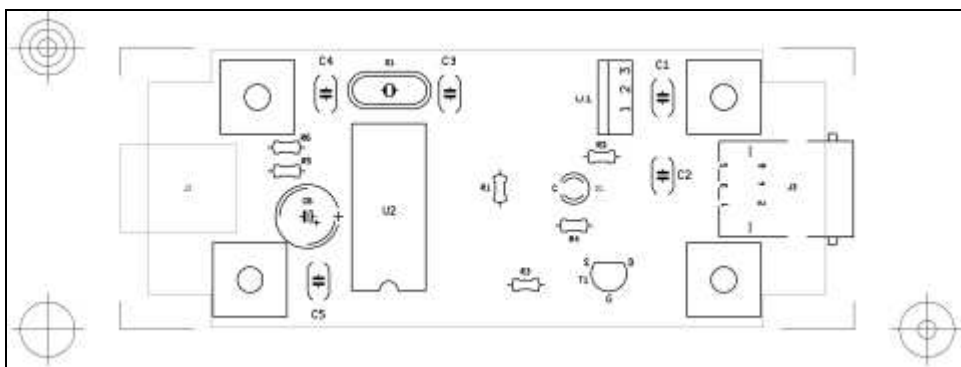
When 0 Volt is set to G(ate) by U2, the P-channel MOS FET (T1) switches on and 5 Volt goes to the LM317 voltage regulator. When 5 volts is set to G(ate), the P-channel MOS FET switches off and 0 volt goes to the voltage regulator (U1).

U1 is an adjustable voltage regulator. With two resistors R1 and R2 the output voltage is set. The formula to determine the resistor values for the output voltage is:

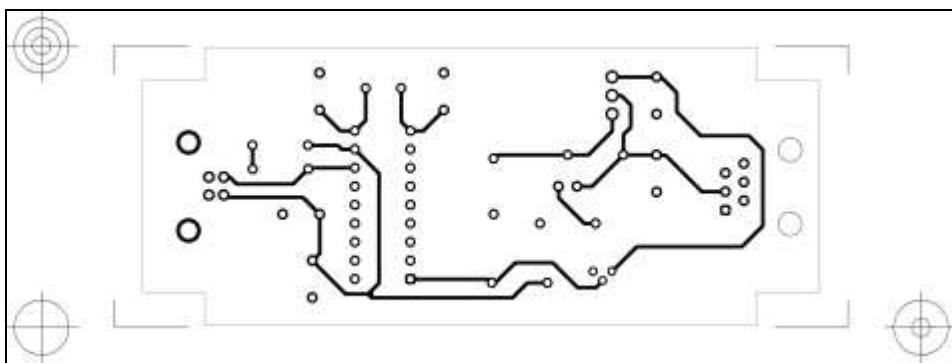
$$V_{out} = 1.25 * (1 + R3/R4). \text{ In this case: } 1.25 * (1 + 680/560) = 2,77 \text{ Volt.}$$

NOTICE: In later models pin 1, 2 and 3 are connected as well as pin 4, 5 and 6, due to power loss. See also the [3.5 Cables](#) section.

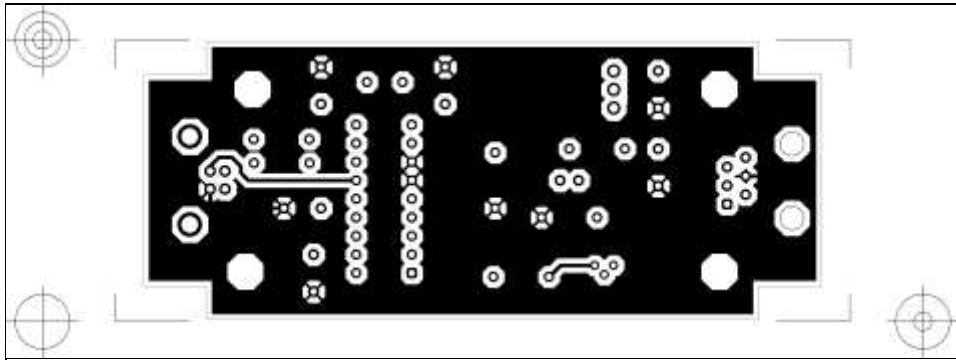
- [Wiiscan\\_pcb.pdf](#) is the printed circuit board and solder mask for the wiiscan\_sch.pdf schematic.



USB\_power\_module\_solder\_mask.png

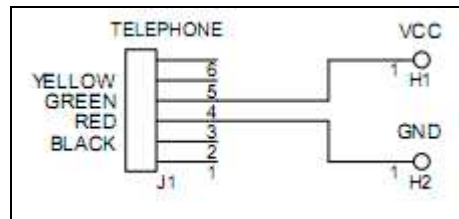


USB\_power\_module\_pcb\_side\_a.png



USB\_power\_module\_pcb\_side\_b.png

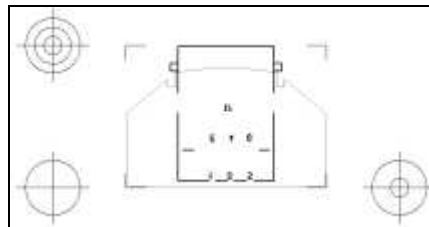
- [Wiimote\\_pwr\\_sch.pdf](#) is the schematic for the power connection of the Wiimote inside the battery compartment. The Wiimote is powered via a plain Cat 5 network cable with two RJ11 (6P6C) connectors. See also section [Cables](#)



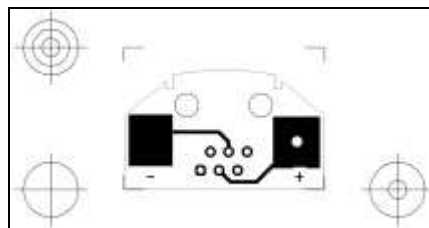
wiimote\_power\_schematic.png

- [Wiimote\\_pwr\\_pcb.pdf](#) is the printed circuit board and solder mask for the powering inside the Wiimote battery compartment. This small printed circuit board serves to hold the clips that power the Wiimote.

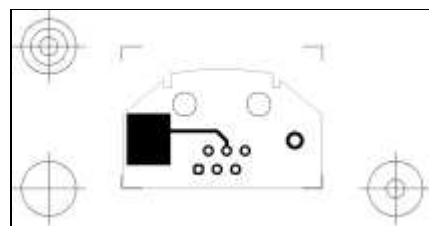
NOTICE: In later models pin 1, 2 and 3 are connected as well as pin 4, 5 and 6, due to power loss. See also the [3.5 Cables](#) section.



wiimote\_power\_solder\_mask.png



wiimote\_power\_pcb\_side\_a.png



wiimote\_power\_pcb\_side\_b.png

NOTICE: In later models pin 1, 2 and 3 are connected as well as pin 4, 5 and 6, due to power loss. See also the [3.5 Cables](#) section.

In the next section you find the electronic parts list for the USB power module and the Wiimote.

## 3.2 Parts lists

### Wiimote

The powering inside the Wiimote has no electronic parts, it just contains:

==== Parts List for Wiimote power +====

J1	1	RJ11, TYPE	.....
H1, H2	2	Battery clips,	TYPE NUMBER ...??

For further information on Registered Jack (RJ) see: [http://en.wikipedia.org/wiki/Registered\\_jack](http://en.wikipedia.org/wiki/Registered_jack)

You also need some glue to fix the RJ11 (6P6C) connector to the battery cover. The place of J1 is rather critical. See picture and measures below.

### USB power module

Below is the parts list for [Wiiscan sch.pdf](#).

==== Parts List for USB power module +====


D1	1	Light Emitting Diode (LED) 3 Volt	
U1	1	LM317	
U2	1	CY7C63723-PC, Delcom Item#: 902370	
T1	1	BS250	
X1	1	6 MHz crystal	
C1	1	4,7 uF	
C1, C2, C5	3	100 nF	
C3, C4	2	22 pF	
C6	1	1uF, 16 Volt	
R1	1	680 Ohm 5 % 1/4 Watt	
R2	1	560 Ohm 5 % 1/4 Watt	
R3	1	10 k Ohm 5 % 1/4 Watt	
R4	1	60 Ohm 5 % 1/4 Watt	
R5	1	6,8 k Ohm 1 % 1/4 Watt	
R6	1	680 Ohm 1 % 1/4 Watt	
J2	1	USB B, Type HDR_4	
J3	1	RJ11, TYPE	.....

ANY REMARKS HERE? NIELS?


## 3.3 Components


Below are the data and layouts of the main components.

**Cable**




**Device**






USB A



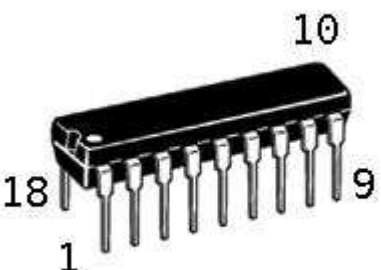
USB B



USB mini

Pin	Signal	Color	Description
1	VCC	<span style="color: red;">■</span>	+5V
2	D-	□	Data -
3	D+	<span style="color: green;">■</span>	Data +
4	GND	■	Ground

USB\_conectors.png

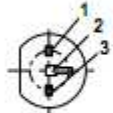
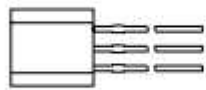
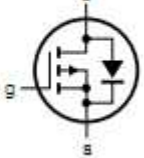


**CY7C63723**  
**18-pin SOIC/PDIP**

P0.0	1	18	P0.4
P0.1	2	17	P0.5
P0.2	3	16	P0.6
P0.3	4	15	P0.7
P1.0	5	14	P1.1
VSS	6	13	D+/SCLK
VPP	7	12	D-/SDATA
VREG/P2.0	8	11	VCC
XTALIN/P2.1	9	10	XTALOUT

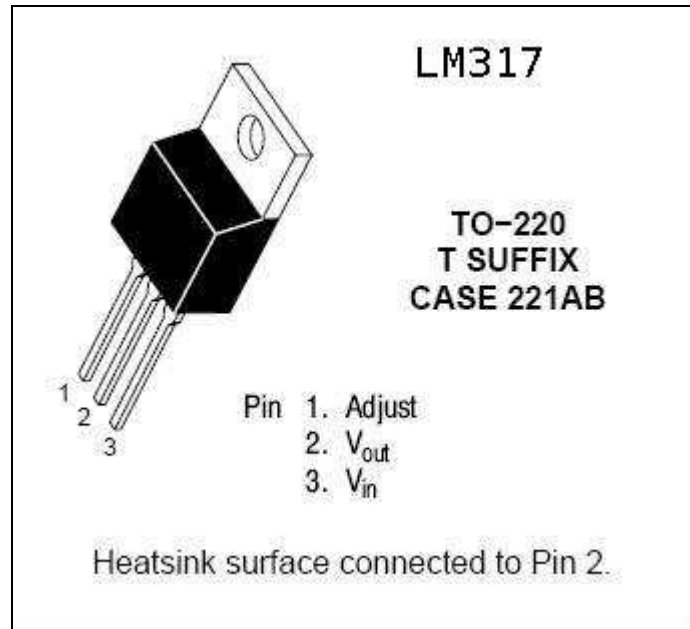
PDIP-18.jpg CY7C63723.png

**1=source, 2=gate, 3=drain**

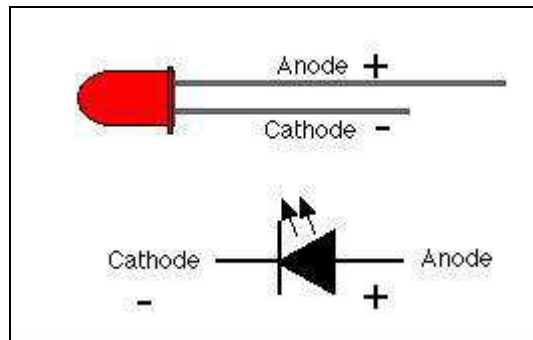




**BS250** MAN147

BS250.png



LM317.jpg



symbol\_led.jpg

### 3.4 The building

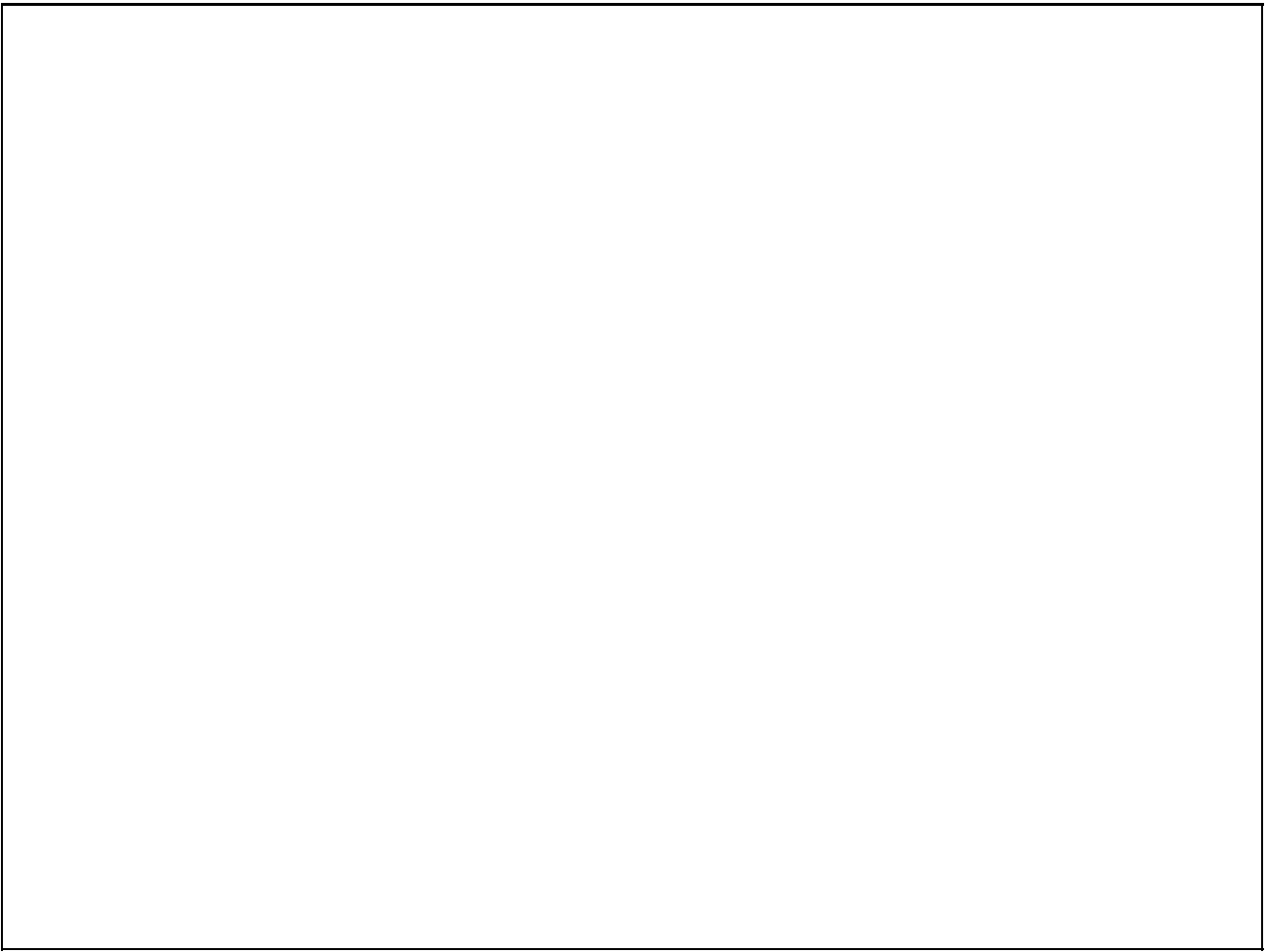
Here are some pictures that may be helpful:  
First, a picture of the DIY-kit.



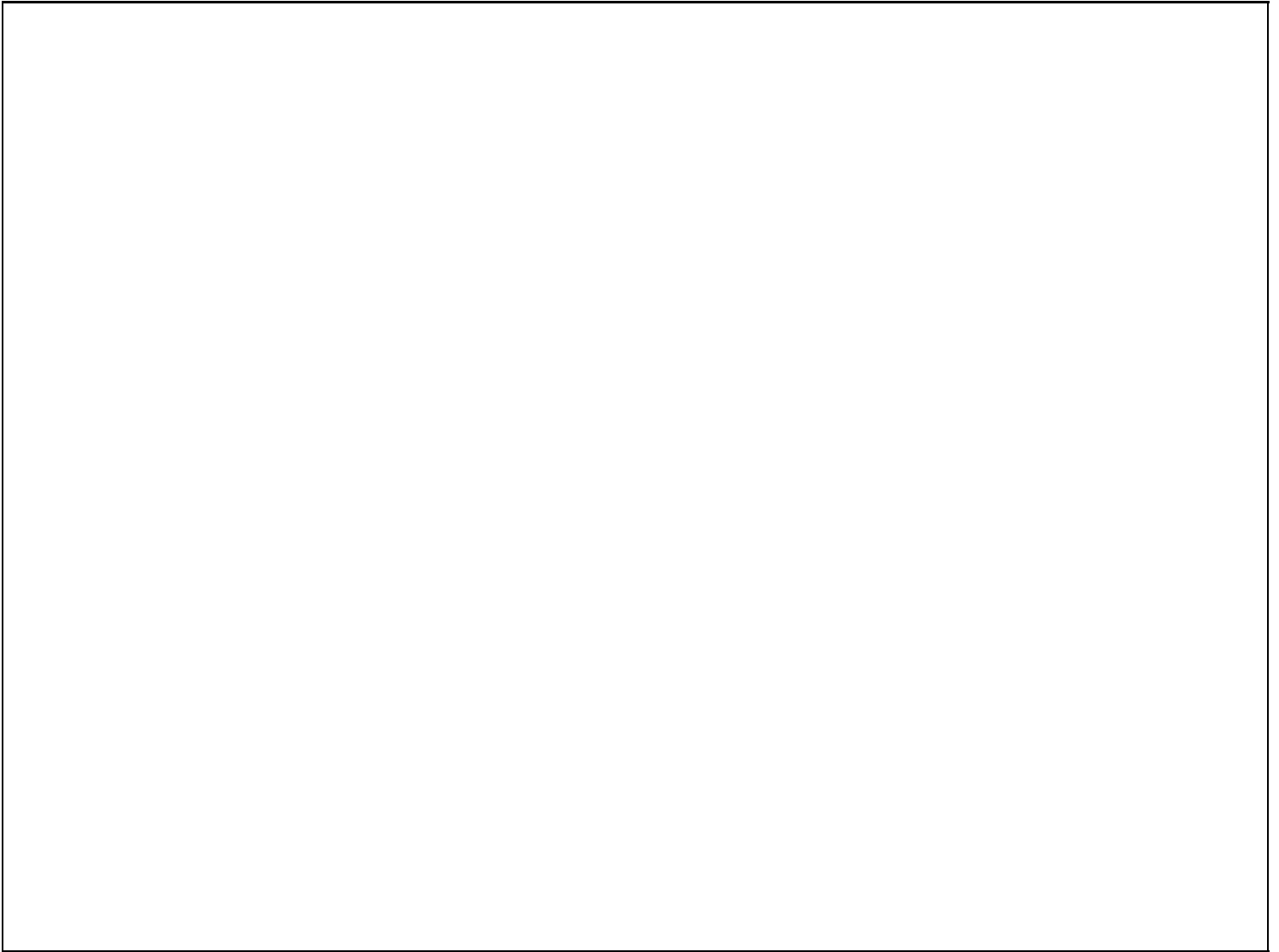
diy-kit.png

A picture of the battery lid of the Wiimote. The place of the holes is rather critical, specially the hole for the cable because its distance is determined by the clips that replace the batteries. The RJ11 hole measures 12x13 millimeter. The hole for the ball head ceiling mount is 7 millimeter. The RJ11 connector is glued to the battery lid.

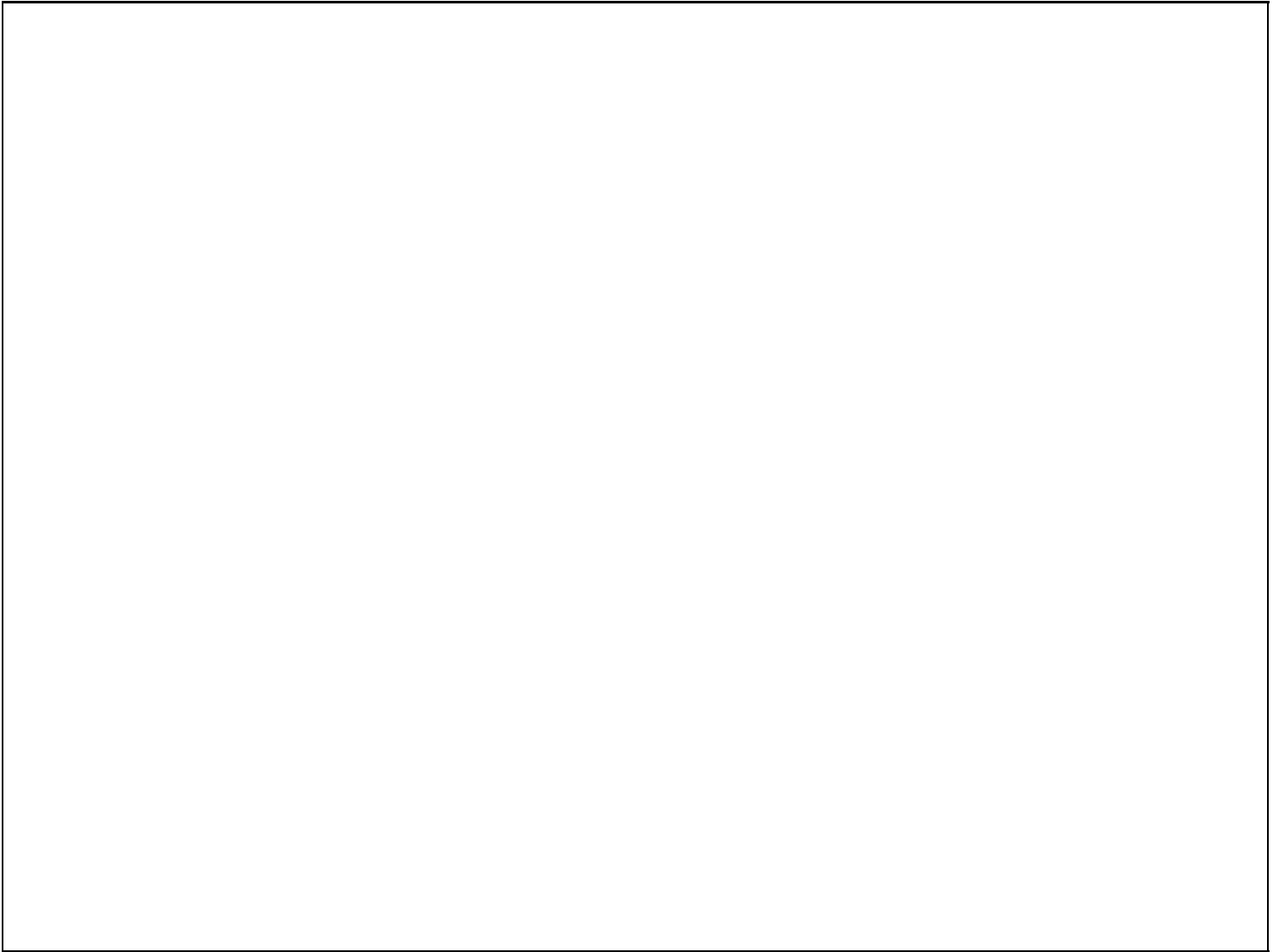




**wiimote\_power.png**

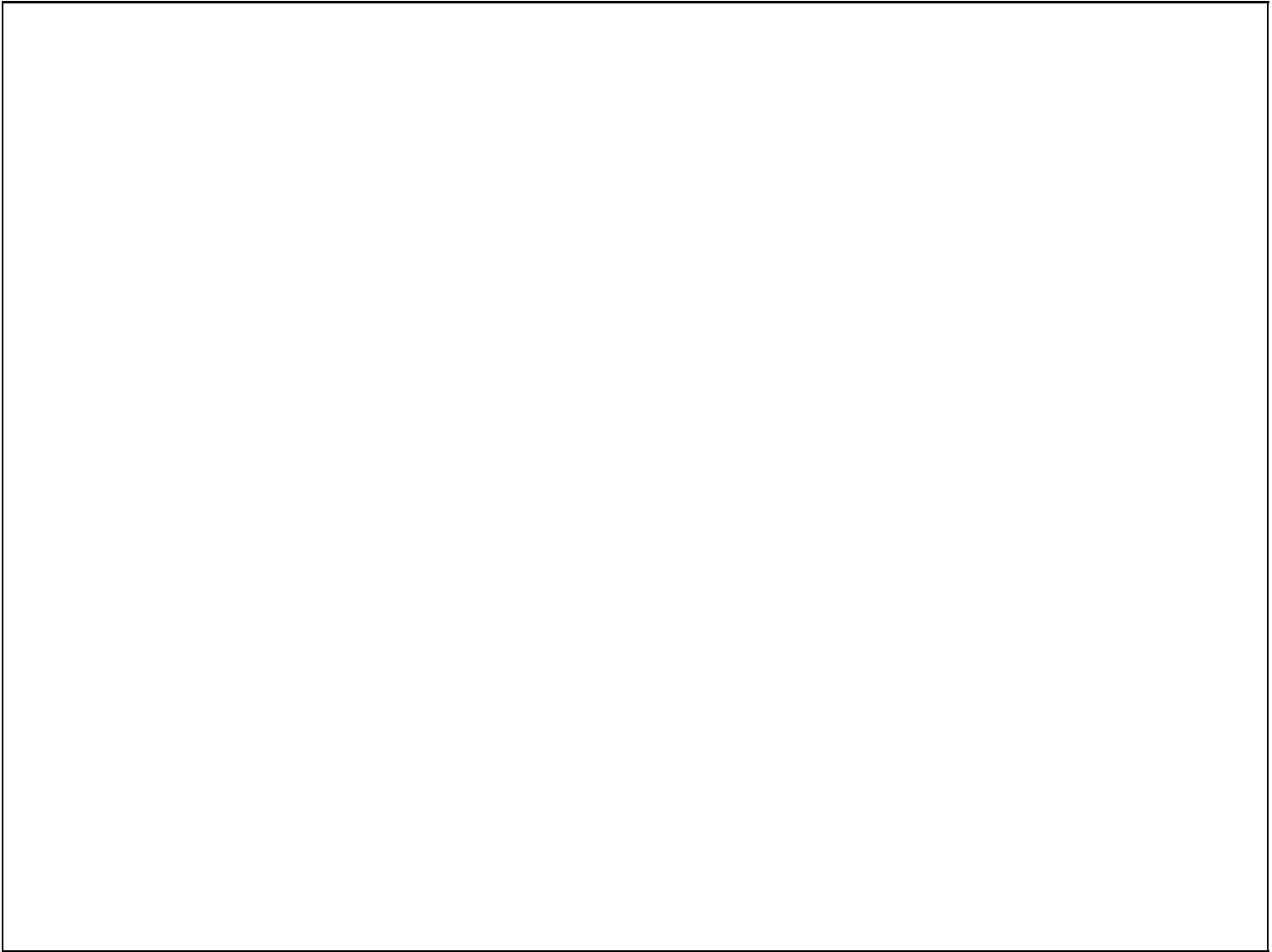


**battery\_lid\_outside.png**



**battery\_lid\_inside.png**

What you have constructed, when built from scratch, could look like:



pcb\_ready.png

Some general advice:

- Do not solder the chip directly on the PC board but use an 18 pins PDIP (Plastic Dual Inline Package) connector.
- Transistors are heat sensitive. Use a pair of tweezers to conduct the heat during soldering.
- Be careful with the BS250. Reversing the source and drain will probably result in a blown up transistor.
- Be patient.

### 3.5 Cables

Two cables are used for this project:

- To feed and send data to the power module, a plain USB2 A B cable is used.
- Powering the Wii is done with two RJ11 (6P6C) connectors and plain Cat 5 networking cable. 2 wires of the cable are not used. Wiring:

RJ11 connector	---Cat 5 network cable---	RJ11 connector
pin 1, 2 and 3		pin 4, 5 and 6
pin 4, 5 and 6		pin 1, 2 and 3

The wire colors are not important, nor is the wiring between 1 and 4 etc. In the Wii and the power module 1, 2 and 3 are connected as well as 4, 5 and 6.

**NOTICE:**

Since 2009 we use plain networking cable with 8 pin RJ 45 connectors.

**3.6 Preliminary tests**

It makes sense to extensively test your work before going any further. We can assure you from self gained experience that a Wiimote will not survive 5 Volt. For the preliminary tests you need a volt meter. Check the following points:

- Is everything corecctly conneced? Are you sure? Please check and double check. It's not a bad idea to have someone else inspect your work. She might have a fresh view on the matter.
- Meticuously inspect the printed circuit board for solder spikes and accidentally connected copper lanes.
- Test the input voltage on the USB side. It shoudl be around 5 Volt.
- Do a simple test. Remember, the BS250 switches the power on/off. When 0 Volt is set to G, the BS250 switches ON and 5 Volt goes to the LM317.  
Bear in mind that D (drain, pin 2) is permanently connected to 5 volt via a 10 K resistor, so the LED should be *off*.
  - Connect the printed circuit board to the PC. Power on the PC. The LED should stay OFF.
  - Connect D (drain, pin 2) to GROUND. The LED should light.
  - When the LED is on, 5 volt goes to the LM317 voltage regulator.
  - With a voltmeter, check the output voltage of the LM317. It should be around 2,77 to 2,85 Volt, depending on the precision of the resistors used.
- If necessary, remove the battery lid from the Wiimote.
- Connect the RJ11 cable between the USB power module and the Wiimote.
- Check if the polarity is correct on the battery clips.
- Unpower the board.
- Before you insert the microcontroller in its socket, ground yourself and your desktop in order to prevent static electricity ruiining the microcontroller.
- Observe the notch on the chip and the notch on the socket and carefully insert the chip in the socket. The pens are fragile.
- Ready!

Close the USB power module enclosure and take a break. After the datasheet section below we will test the USB power module with the Wiiscan software.

**3.7 Datasheets**

[wiihardware.pdf](#). This is Carsen Frigaards description of his USB power module schematic. One might wonder why we did not simply copy his schematic. The reasons is that we wanted all parts to be as simple as possible and enlarge the availability of parts. One part was difficult to get in the Neterhlands, so we expected similar problems elewhere.

- [CY7C637xx.pdf](#). Datasheet for the CY7C637xx microcontroller

- [BS250.pdf](#). Datasheet for the BS250 P-Channel MOS FET

- [LMx17.pdf](#) Datasheet for the LM317 voltage regulator.

[\(top\)](#)

**4 Testing the USB powre module**

Now that the USB power module and the Wiimote powering are ready, the next step is to test them. We can only test the USB power module by installing a driver (optional, see below) and editing a file. Both procedures are described below. The Wiiscan software is already downloaded and installed on the PC.

## 4.1 Determine the microcontroller version

Before we can use the USB power module and Wiiscan, we have to determine which Delcom microcontroller version is used in the USB power module. Is unlikely that it's a Delcom 'Generation 1' with part number 802370. If it is the case, then it will need a driver.

When a Delcom 'Generation 2' with part number 902370, it is a 'Generation 2' micro controller which does not need a driver.

The test procedure to check the version of the microcontroller is as follows:

- Switch on the PC and log in with administrator rights.
- Choose a USB 2.0 port. Choose this port carefully and, once chosen, do not change it anymore. An installed driver 'remembers' the port.
- With a felt marker, mark the port 'Wii'.
- Insert the USB power module USB cable into the PC. As said, two conditions can occur:
- **Generation 2:** When inserting the USB cable from the USB power module in the PC, you just hear the 'Dong dong' sound. The new hardware is automatically detected. This is the normal condition. In the power module is a Delcom 'Generation 2' microcontroller with Delcom part number 902370. Skip step 2 below and proceed with the next one: 'Testing the USB power module'.
- **Generation 1:** When inserting the USB cable from the USB power module in the PC, you get a text balloon 'Found New Hardware' and the window 'Found New Hardware Wizard' opens, asking you to install the driver.  
So it's the 'Generation 1' micro controller, Delcom part no. 802370 is in the USB power module. This chip works us just as perfect but needs a driver. This is no problem. Please proceed to the next section.

## 4.2 Installing the driver

The driver is in `C:\wiiscan\redist\`. To install the driver, proceed as follows:

1. We assume you are still on the window: New hardware found. Check the 'No, not this time' option and click [Next].
2. In the next window, you check 'Install from a list or specific location (Advanced)'. Click [Next].
3. In the next window, the option 'Search for the best driver in these locations' is already checked. Also check 'Include this location in the search'.
4. The [Browse] button become selectable. Click [Browse].
5. The 'Browse for Folder' window opens. Navigate to `C:\wiiscan\redist\`.
6. Click 'redist'. The [OK] button becomes visible and is clickable.
7. Click the [OK] button. The window closes.
8. Window: Found New Hardware Wizard: the location is specified: `C:\wiiscan\redist`. Click [Next].
9. Window: Wizard new hardware found: Hardware Installation: the 'Delcom USB IO Driver' is found.
10. Also the Window 'Hardware Installation' opens. You get a warning that the driver does not meet the Windows Logo specification. Ignore the warnings to end the installation and click [Continue Anyway]. The window closes.
11. Window: Found New Hardware: The installation of the software for the following device is finished. Delcom USB IO driver. Click [Finish].
12. You get a text balloon 'Found New Hardware'. The new hardware is installed and can be used.
13. As usual, reboot.

Check your installation by going: Start > Settings > Control Panel > System > W: System Properties > T: Hardware > B: Device Manager > Universal Serial Bus Controllers > +. You should find a line with 'Delcom USB IO Driver'. The driver for your USB power module is successfully installed.

NOTICE: When you have the normal distribution of the USB power module, you have a Delcom chip with part No. 902370. This 'Generation 2' chip needs no driver and installation is not necessary.

### 4.3 Editing wiiscan.ini: power mode

Before you can test the USB power module, you have to initialize wiiscan to use the Delcom microcontroller. This is done by editing the wiiscan initialization file. The file is located in `C:\wiiscan\bin\Release` and it needs 'option 3'. Use Notepad to edit the initialization file. Browse through the file to get familiar with it. The `%` sign indicates that the line is either a comment, or an option that is commented out, i.e. not working.

Proceed as follows to edit `wiiscan.ini`:

- Open Notepad and navigate to `C:\wiiscan\bin\Release\`. Open the `wiiscan.ini` file.
- Search for the lines:

```
% 0=no power cycle, 1=USB hub control, 2=USBm IO, 3=USB Devcon
option_usbpowermode=0
```

NOTICE: TYPO: 'Devcom' should be 'Delcom'.

- In the line `option_usbpowermode=0`, change the 0 (zero) into 3 (three). Save your work.
- Check your work by opening the file again and look for:

```
% 0=no power cycle, 1=USB hub control, 2=USBm IO, 3=USB Devcon
option_usbpowermode=3
```

Now we can do the first test on the USB power module.

### 4.4 Testing

In the previous steps the version of the Delcom microcontroller was established and eventually a driver was installed. Also `wiiscan.ini` was edited to use the Delcom microcontroller.

Now we can test the USB power module. Proceed as follows:

- Power up the PC and login with administrator rights.
- Insert the USB power module, via the USB cable, into the chosen USB slot on the PC.
- You hear the sound 'Dong, ding', indicating that the device is found.
- Open a DOS box and `cd` to `C:\wiiscan\bin\Release\`.
- With the '`cd`' command, check if you are in the right directory.
- On the command line type: `wiiscan -usbup`
- The blue LED inside the USB power module should light up.
- On the command line type: `wiiscan -usbdown`
- The LED is switched off.

Great! The USB power module functions correctly. If it's not working, go to the next section.

### 4.5 Problems

What to do when the LED does not light. There is little help we can offer here. Just a few hints and questions in the hope it helps:

- LED polarity wrong?
- Broken LED
- Check wiring again.
- See also the next section [5.4 Problems](#)

[\(top\)](#)

## 5. Testing the Wiimote

In a previous section the USB power module was tested. In this section the Wiimote is attached and tested. To perform the test we have to tell Wiiscan which Wiimote to use by editing the MAC (Media Access Code) address in wiiscan.ini. You already noted the MAC address on a piece of paper when you were in section USB Bluetooth dongle test

### 5.1 Editing wiiscan.ini: MAC address

To enter the MAC address in wiiscan.ini, proceed as follows:

1. Open the wiiscan.ini file again with Notepad and look for the lines:

```
% a list of allowed wiimotes in the form xx:xx:xx:xx:xx:xx, wildcard match is
00:00:00:00:00:00
allowed_wiimote_adr=00:00:00:00:00:00
```

2. In the line allowed\_wiimote\_adr=00:00:00:00:00:00, replace the string '00:00:00:00:00:00' with the MAC address of your Wiimote, for example: 00:21:BD:A4:07:AD. Use uppercase letters!  
Save the file.
3. Check your work. Open the file again and look for allowed\_wiimote\_adr=00:21:BD:A4:07:AD

You are now ready to connect the Wiimote to the USB power module.

### 5.2 Connecting

If necessary, remove the batteries, replace the lid and connect the Wiimote to the USB power device with the cable with the two RJ11 6P6C connectors. At this point you have everything connected. The Wii has its buttons 1 and 2 continuously pressed with a piece of plastic and a tie-wrap. See section [12.. Wiimote](#) for details.

A permanent solution might be a piece of perspex as on this picture.





perspex\_wii\_permanent\_on.png

### 5.3 Testing

The basic test is the same as with the USB power module. We assume you have done the previous test and the USB power module functions.

1. Open a DOS box and cd to C:\wiiscan\bin\Release\.
2. On the command line type: wiiscan -usbup
3. The 4 blue LEDs on the Wiimote should blink.
4. On the command line type: wiiscan -usbdown
5. The 4 LEDs should go off.

Great! Both the USB power module and the Wiimote get power. We now can make use of all the fine features of Wiiscan.

### 5.4 Problems

Reported problems:

- **LED on USB power module lights up but the LED's on the Wii stay off**  
 . Is the cable really working? Carefully open the battery lid on the Wii. Prevent bending the battery clips.. With a volt meter, check if power is on the battery clips when the USB power LED is on. You should measure something between 2,8 and 3 volt. If this is indeed the case, the clips do not make contact to wii. Inspect and if necessary, carefully bend the clips.  
 Close the cover by laying the cover on the Wii in such a way that the front side and the back side lay almost flat on the Wii, but *without* the two tips on the front side of the lid in their holes. Shift the lid backward, let the clips slide in their holes and carefully press the backside of the lid, thus closing the cover.
- **Broken cable/connections?.** Are the connectors well crimped? Check cable.

- **Typo in MAC address** Thanks to Kurt! The typo results in: you can switch on the LED in the power module (with wiiscan -usbup), but the 4 LED's on the Wii do not blink. Remember that the letters in the MAC address must be in UPPERCASE.

[\(top\)](#)

## 6. Wiiscan

In the previous two sections the USB power module and the Wiimote were tested for powering capabilities. We now take up the final steps: connecting and fine tuning. First we discuss some files, and finally a small section about problems.

### 6.1 WIISCAN\_manual

This is the best starting point, in other words RTFM (Read the Fine Manual)

```
wiiscan(1) wiiscan(1)
```

#### NAME

wiiscan - a connection utility for wii console remotes

#### SYNOPSIS

```
wiiscan <-a <device> | -c <device> | -d <device> | -r | -s | -usbup |
-usbdown> > [-cf <file>] [-lf <file>] [-b <sleep>] [-t <sleep>] [-u
<sleep>] [-p <sleep>] [-w <sleep>] [-q <usbradio>] [-f <removemode>]
[-m <powermode>] [-l <wiilibrary>] [-wb] [-v]
```

#### OVERVIEW

wiiscan is a canvas function for a number of different scanning and connection utilities. It can detect build-in bluetooth radios, scan for nearby bluetooth devices, connect to a specific device and remove that device again from the hardware.

The main feature of wiiscan is to automatically connect to a wii remote (wiimote). This can be quiet cumbersome on a Windows system, and the nessecary steps for doing a robust, working connection is done by

Delete wiimote hardware HID bluetooth entry:

delete old entries of the wiimote in the bluetooth hardware. On some windows system the wiimote is readily detected, after the first manually installation. Pressing the "1-2" makes the wiimote discoverable. On other systems (including mine), any attempt to connect to the wiimote, after one successful or unsuccessful connection attempt, will always fail! Removing the HID and wiimote registry entries before the next attempt cures this feature.

Cycle USB bus:

the wiimote can be switched automatically to discoverable mode, if the power is briefly cut from the device. This can be done on a wiimote powered by the USB +5 volt (with a proper voltage regulator to bring it under +3 volt). This step hence turns the USB hub off, killing the power, and turns it on again. The "1-2" buttons on the wiimote must be pressed at all times, done by say some tape or gaffa!.

Scan for wiimote:

now comes the main part of the connection; scanning for a

trying to open the wiimote and reading some data from it. All steps involve a time variable, meaning that say installing the HID causes some windows registry fiddling, that takes a rather variable amount of time, and the next step is critically dependent on the former step to be finished. Hence a number of tunable waiting variable is introduced, Important is to being able to reach the final connection step before the wiimote goes out of the discoverable mode and automatically turns off.

A note on USB voltage cycling:

This software solution is able to restart the wiimote in discoverable mode automatically. But this require that the wiimote is powered by a USB cable to the PC and that the "1-2" button combination is permanently pressed. Cycling the power for the wiimote with the "1-2" buttons pressed enables the discoverable mode, and the continuously pressing "1-2" does not interfere with wiimote operations hereafter.

So this is a non-intrusive fix to the so called "Ladder problem" ([http://wyxs.net/web/wiimote/digital\\_whiteboard.html](http://wyxs.net/web/wiimote/digital_whiteboard.html)).

Power control over USB hubs is dependent on the particular hub devices, and disabling the power may not be possible for a single USB port only. wiiscan uses a trick of disabling all USB hubs and then only enable and disable a single hub.

Disabling all hubs is a precondition to running wiiscan and it can be accomplished by either going into "Control Panel | System | Hardware | Device manager", and manually disable all USB hubs or by using "devcon", say (careful, these commands applies only to my system)

```
devcon          disable          "@PCIN_8086&DEV_293*&SUB-
SYS_20F117AA&REV_033&B1BFB6*"
```

```
devcon          disable          "@PCIN_8086&DEV_293*&SUB-
SYS_20F017AA&REV_033&B1BFB6*"
```

being careful that the pattern matches only our USB hubs!. You can test this by

```
devcon          status          "@PCIN_8086&DEV_293*&SUB-
SYS_20F117AA&REV_033&B1BFB6*"
```

```
devcon          status          "@PCIN_8086&DEV_293*&SUB-
SYS_20F017AA&REV_033&B1BFB6*"
```

A single USB now may be enabled or disabled by

```
devcon disable @PCIN_8086*DEV_2934*SUBSYS_20F017AA*
```

again with the particular address for your system as a variable you need to lookup. The voltage on the USB bus can now be checked Using a voltmeter (see details in [http://wyxs.net/web/wiimote/digital\\_whiteboard.html](http://wyxs.net/web/wiimote/digital_whiteboard.html)).

NOTE: The configuration file of wiiscan contains the particular system dependent USB hub pattern matches.

NOTE: be careful of connecting other devices to the USB bus, since the power cycle may cause severe interfere with the device.

- c <device>  
connects to a device.
  
- d <device>  
deletes a device, clears HID and bluetooth registry entries.
  
- r  
looks for active internal bluetooth radio devices.
  
- s  
scans for external bluetooth devices.
  
- usbup  
turn the USB hub on.
  
- usbdown  
turn the USB hub off.

Default mode: wiiscan -a " Nintendo RVL-CNT-01"

Note: "nintendo" is a shortcut for "Nintendo RVL-CNT-01"

#### OPTIONS

- cf <file>  
load a specific configuration file.
  
- lf <file>  
specify a distinct logfile.
  
- b <sleep>  
auto-mode bluetooth connection sleep in milliseconds.
  
- t <sleep>  
bluetooth scanning interval in milliseconds.
  
- u <sleep>  
auto-mode USB connection sleep in milliseconds.
  
- p <sleep>  
automode usbm post-connection sleep in milliseconds.
  
- w <sleep>  
timeout for wiimote in milliseconds.
  
- q <usbradio>  
use bluetooth radio with this address. Note, this functionality is not working yet.
  
- f <removemode>  
pre-remove mode of device, 0=remove if not connectable, 1=always remove, 2=never remove

```
-l <wiilibrary>
    use specific wiimote library, lib can be one-of
    (wiiuse,wiimotelib).

-y      scan retries in automode.

-wb     start whiteboard in auto-mode

-v      enable extra debugging printouts
```

#### FILES

wiiscan looks for a file names wiiscan.ini when executing in the auto-mode. See detail in the file.

#### TIMING

Various timing parametes can be set on the command line or in the inifile. The process of connecting is

1: USB power down

2: Delete old HID entries

3: Sleep at least option\_usbsleep (from 1) before commencing

4: USB power up

5: Sleep at least option\_usbmsleep (from 3) before commencing. Sleep is only nessecary for USBmicro devices, since step 4 is much faster using this device compared to the USB hub.

6: Find and wiimote and connect. Scan bluetooth the duration in option\_timeout.

7: Sleep option\_btssleep before commencing; waiting for HID entries and new-hardware-found to finish in windows.

8: Connect to the wiimote, Scan for wiimotes using the duration option\_wiitimeout.

#### EXAMPLE

Scanning for devices nearby:

```
wiiscan -s
```

Auto-connect to a nintendo device, scan bluetooth for four seconds, verbose on, and enable start of whiteboard software after a successful connection

```
wiiscan -a nintendo -t 4000 -v -wb
```

Cycle USB bus voltage

## COMPILING

The source code can be compiled with MS Visual C++ Express 2008 (<http://www.microsoft.com/express/vc/>) or similar. It also needs wiiuse dlls (<http://www.wiiuse.net/>). If wiiuse is to be compiled by it self it needs Windows SDK and DDK, but running wiiscan with just the wiiuse binaries is the easiest option.

It does not work on Windows 2000, and has not been tried out on a Vista system.

## TESTED SYSTEMS

The wiiscan has been tested on a Lenovo Thinkpad R500, XP professional (without build-in bluetooth) with a Trendnet TBW-102UB bluetooth dongle, and on a ASUS eeePC 1000H with XP home.

Only the MS bluetooth stack was tested.

### Lenovo setup

Windows XP professional, version 2002, SP2 USB dongle: Trendnet TBW-102UB bluetooth(Broadcom Ultimate Low Cost Bluetooth 2.0+EDR USB), date 24-02-2004, driver 5.1.2535.0 Microsoft BT stack: date 03-08-2004, driver 5.1.2600.2180

### eee setup

Windows XP home, version 2002, SP3 USB dongle: buildin Azware BT252, date 13-04-2008, driver 5.1.2600.5512 Microsoft BT stack: date 13-04-2008, driver 5.1.2600.5512

## BUGS

### 1: restart pop-up (FIXED)

Installing new hardware causes windows to require restart. Happens once in a while, balloon pop-ups reports hardware, that where installed but not working properly. A restart pop-up wants to reboot the PC. Small fix: just delete the device and re-run "wiiscan -c nintendo".

FIX: the 'Change of systemsettings ... Do you want to reboot now?' popup, happens only when the program runs for longer than 19 seconds. Keeping the connection time within 19 sec seems to cure the problem, Adding a sleep at the end of the program, say Sleep(20000) will eventually bring up the dialog again. The dialog can anyway safely be ignored!

### 2: discoverable mode fast shutdown (FIXED)

Sometimes the wiimote goes quickly out of discoverable mode, it takes it only about 3 seconds from turn-on to turn-off. This makes it hard to obtain a connection to it. Both my wiimotes does this once in a while, after failed connection attempts.

Pressing one button only "1" or "2" makes the wiimote blink for a short time, but it is not really discoverable.

FIX: Initialization of wiilib has been rewritten, making it recall the HID initialization routine. Code for testing the wiimote connection has also been introduced.

### 3: buttons not working (OPEN)

Pressing the 1-2 button combination sometimes fails to turn-on

## 4: radio null (OPEN)

Sometimes the BT radio fails to reinitialize after a USB down/up flip, this means that "RadioInfo(timeout,true,dbg)"return NULL. Can be fixed by placing the bluetooth radio device on another bus than the USB.

## 5: keep blinking (OPEN)

Sometimes the wiimote is found and connected OK, but the LEDs keeps blinking (normal connect mode: LEDs are turned permanently on). This does however not affect connectability, and the wiimote does not turn off again automatically.

## 6: failed to find wiimote (OPEN)

Wiimote failed to find devices. This may be a non-fatal error or an error caused by an undervolted wiimote. The "wiimote\_find(0x0,4,2/4/6)" keeps returning 0.

## 7: remove failed (OPEN)

Sometimes the remove steps fails, but this may be non-fatal

```
Removing device <Nintendo RVL-CNT-01>
** error: failed to find device
Done [FAILED]
```

## 8: balloon-tips (FIXED)

Balloon-tips are annoying when connecting new hardware. Small fix: do

Windows Registry Editor Version 5.00

```
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer1]
"EnableBalloonTips"=dword:00000000
```

## 9: double delete (FIXED)

Double delete of nintendo device may cause BluetoothFindFirstDevice() to return null Fixed by removing the throw, replacing it with a "if (hbf()==NULL) then return false"

## 10: BSoD (OPEN)

The "Blue Screen of Death" was encountered a number of times, indicating a erroneous device driver. The cause may be in the MS bluetooth stack or in the wiimote lib. The BSoD mainly occurred in the first phase of this project and havent been seen for a while with the current version (v0.7).

Only happens for MS bluetooth stack version "Microsoft BT stack: date 03-08-2004, driver 5.1.2600.2180" (Lenovo version).

## 11: New Hardware found wizard (OPEN)

Sometimes the wizard appears out of the blue, when deleting the HID entry and trying to reinstall it. It is the call BluetoothSetServiceState(...) that messes the Window system up. Disabling the wizard or disabling the Plug and play system do not seem to be an option, since the HID then newer get installed, and ends up in a failty state. A reboot of the system does not cure this defect.

disable or enable the USB controller fails. A status on the USB device gives a strange result.

FIX: reboot the machine.

13: Open Device fails to find a Nintendo (PARTLY FIXED)

When opening a named nintendo device, say 'Nintendo RVL-CNT-01', the bluetooth fails to get the name from the device. This results in an empty name, and hence matching on the name fails.

FIX: use a device address instead of a name. Put, say 'allowed\_wiimote\_addr=00:19:FD:CC:60:61 00:19:1D:D6:65:E5' in the ini file.

14: Wiiscan fails to run (FIXED)

Running wiiscan in a dosbox makes it terminate immediately. Running the tray version, or under MS VS Express causes it to display an missing DLL dialog (msvc90.dll and friends).

FIX: this is a MS bug, but can be fixed by setting the MT library in the project setting to use static libraries instead of dynamic DLLs.

15: BluetoothFindFirstDevice() stalls (FIXED)

The BluetoothFindFirstDevice() sometimes never returns. Happens only for MS stack version "Microsoft BT stack: date 13-04-2008, driver 5.1.2600.5512" (see stack).

Code stalls here:

```
DeviceAutoClose<HBLUETOOTH_DEVICE_FIND,BOOL> hbf(BluetoothFindFirstDevice(&bdsp,&bdi),&BluetoothFindDeviceClose);
```

FIX: start the BluetoothFindFirstDevice() function in a thread, terminate the thread if it has run for longer than, say 2+time-out.

16: Wiimote drops connection after scanning (PARTLY FIXED)

If a connection attempt goes well until the last part, it is most likely due to low-power batteries. The final stage of connecting draw extra power, that might cause the wiimote to shut-down.

SEE ALSO

wiiuse Wiiuse is a library written in C that connects with several Nintendo Wii remotes. Supports motion sensing, IR tracking, nunchuk, classic controller, and the Guitar Hero 3 controller. Single threaded and nonblocking makes a light weight and clean API.

Licensed under GNU GPLv3 and GNU LGPLv3 (non-commercial) by Michael Laforest,

(<http://www.wiiuse.net/>)

Wiimote Whiteboard

Whiteboard software by Johnny Chung Lee.



```
devcon(1)
```

USB management software.

(<http://support.microsoft.com/kb/311272>)

#### VERSION

Version 0.9 NDEBUG

#### AUTHOR

Carsten Frigaard, Mergeit ApS, Kongsvang Allé 37, DK-8000 Århus C,  
[www.mergit.dk](http://www.mergit.dk)

#### COPYRIGHT

Copyright © 2009 MergeIt, Aps. License LGPL3 : GNU lesser GPL, version 3, <<http://www.gnu.org/licenses/lgpl.txt>>. This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

26 Mar 2009

wiiscan(1)

## 6.2 WIISCAN -H

Below is the output of

```
wiiscan -h
```

on the command line:

```

C:\WIISCAN\BIN\RELEASE>wiiscan -h
Usage: WIISCAN <-a | -c | -d | -r | -s | -usbup
| -usbdown> [-cf ] [-lf ] [-b ] [-t ] [-u ] [-p ] [-w ] [-q ] [-f ]
[-m ] [-l ] [-y] [-wb] [-v]
VERSION: 0.9 NDEBUG 32BIT LITENDIAN
modes:
  -a : autoconnect to device
  -c : connect the device, that matches this name
  -d : deletes the device, that matches this name
  -r: lookup and list bluetooth radio devices
  -s: scan external bluetooth devices
  -usbdown: disable usb hubs
  -usbup: enable usb hubs
options:
  -cf : specify a distinct configurationfile, default=wiiscan.ini
  -lf : specify a distinct logfile, default=cout
  -b : automode bluetooth connection sleep in milliseconds,
      default=2000
  -t : timeout for bluetooth stack in milliseconds, default=2000
  -u : automode usb connection sleep in milliseconds, default=500
  -p : automode usbm post-connection sleep in milliseconds,
      default=3000
  -w : timeout for wiimote in milliseconds, default=2000
  -q : use bluetooth radio with this address (not working),
      default=any device
  -f : pre-remove mode of device, 0=remove if not connectable,
      1=always remove, 2=never remove, default=0
  -m : choose USB powercycle mode, 0=no power cycle, 1=use USB
      hub. 2=use USBm IO hardware. 3=use USB Delcon IO hardware

```

```
-nowb: do not start whiteboard in automode
-v: enable extra debugging printouts
default mode: -a "Nintendo RVL-CNT-01"
note: "nintendo" is a shortcut for "Nintendo RVL-CNT-01"
```

```
C:\WIISCAN\BIN\RELEASE>
```

### 6.3 Wiiscan.ini

For lay out in a web page we show the content of wiiscan.ini slightly modiefied. The % sign indicates a comment line; it's not code.

```
% Configfile_begin
    % config file for wiiscan

    % all usb device ids, not used yet in the scanning
        all_usb_hubs=""

    % single power controlling usb hub, only nessesary when powerering
% over usb, for automatic resetting. If not used, put ""
    % active_usb_hub="@PCI\VEN_etc.etc._03\3&B1BFB68&0&E8"
        active_usb_hub=""

    % a list of allowed wiimotes in the form xx:xx:xx:xx:xx:xx,
    % wildcard match is 00:00:00:00:00:00
        allowed_wiimote_adr=00:00:00:00:00:00
    % my set of wiimotes
        %allowed_wiimote_adr=00:19:?D:??:6?:?? 00:1F:C5:??:??:??

    % launching exe file have a problem with spaces, hence renamed exe,
% replaced space with underscores
    % The original Johnny Chung Lee wb software version 0.3, downloaded
% from [http://johnnylee.net/projects/wii/]
        % whiteboard_software="d:/WiimoteWhiteboard/WiimoteWhiteboard_v0.3.exe"

    % same stuff, but binary build from sources and redistributed in this
% package
        %whiteboard_software="WiimoteWhiteboard.exe"

    % and finally, a custom build v0.3 version, with fixed
    % calibration.dat write
        whiteboard_software=WiimoteWhiteboard_v0.3.exe

    % Boon Jin's wb software version 0.6
        %whiteboard_software="d:/wsb/WiimoteSmoothboard.exe"

    % wiiscan parameters
        %option_device="Nintendo RVL-CNT-01"

    % number of bt scanning attempts in a autoscan, timout is
% increased for every scan as retryattempt*option_timeout
        %option_scanretries=3

    % long time (4000 and above, or increase option_scanretries) when
% matching name, short (1000-2000) when matching address
        option_timeout=4000

    % wiimote scanning time, only applicabile used for wiilib
        option_wiitimeout=1000

    % delay bewteen powerdown/up, see man for details
        option_usbsleep=1000
        option_usbmsleep=1000
        %option_btssleep=2000
```

```

option_startwhiteboard=1

% logging output file, typical wiiscan_log.txt , put nothing if no
% logging is wanted
option_logfile=

% wiimote connection library, use none, wiiuse, or wiimotelib
option_wiilib=wiimotelib

% 0=no power cycle, 1=USB hub control, 2=USBm IO, 3=USB Devcon
option_usbpowermode=0

% 0=remove if not connectable, 1=always remove, 2=never remove
%option_forceremove=0

% bt radio to use, not working yet
%option_btradio=00:50:B6:A0:48:8C

% wiiscantray options
% number of connection attempts before autoscan i restarted
option_tray_pool_retries=2

% retry sleep, keep it low (<1000) to catch an open connection
option_tray_sleep_between_pool_retries=250

% sleep bewteen polls, when wiimote is succesful connected, keep it
% hight (>2000) to avoid to much CPU usage
option_tray_sleep_between_pool_loops=1000

% final connection step requires some sleep before windows PnP is ready
option_sleep_before_final_connect=1500

% delay after launch of wb software, keep it at a medium value
% (2000-10000)
option_sleep_after_startwhiteboard=4000
% Configfile_end

```

When editing this file it is recommended to write slashes the Unix way i.e. '/' (omit quotes).

## 6.4 INSTALL.rtf

We simply follow Carstens INSTALL.rtf in C:\wiiscan\bin\Release. We added notes in the text between square brackets.

### INSTALLATION

Version: 0.9

Procedure for wiiscan or wiiscantray:

Prerequisites

- 1a: unzip the wiiscan zip file(s).[1]
- 1b: if you are not running a Microsoft Bluetooth stack, then change it from Widcomm, Bluesoleil or whatever you are using. See the file CHANGINGBTSTACK.rtf.[2]

Running

- 1a: GUI version, run "wiiscantray.exe" in bin/Release or bin/Debug directory.[3]

- 1b: GUI version, run "wiiscan.exe" in

Tune scan speed by giving wiiscan your specific wiimote address. Run "wiiscan -s nintendo" from a prompt to find the address, or use the windows system setting.[5]

Replace the "allowed\_wiimote\_adr=00:00:00:00:00:00" by our specific address in the wiiscan.ini file, and lower the value of "option\_timeout" to 1000 or 2000.[6]

Avoid annoying baloon tips by running: "popup\_disable.reg" found in src/wiiscan-0.9.zip, under Tools.[7]

Avoid annoying crash reports from the Wiimote library by running: "no\_error\_report.reg" found the same place as above.[8]

ENJOY

Mar 26 2009

Carsten Frigaard,  
MergeIt ApS, Kongsvang Allé 37, DK-8000  
Århus C, www.mergit.dk

[1] Done in the Software section

[2] Done in the Bluetooth section

[3] This is the ideal mode for teachers. Make a shortcut in the 'Start Menu' and the teacher only has to power the PC to start the digital interactive whiteboard.

Remember, we did not use the 'debug' directory, so use C:\wiiscan\bin\Release for the shortcut

[4] See section 'Wiiscantray' below.

[5] Done in the Bluetooth section

[6] That's not a problem. You have already edited the Wiiscan.ini file. Do this as last step when you know everything is working fine.

[7] That's a matter of opinion. It could also serve as a assurance for the teacher that her beloved whiteboard is functioning (or not). Another good reason to not (yet) edit this option is that you cannot test Bluetooth functionality well enough.

[8] This is a rather drastic measure. First try a more specific and reversible way as described in <http://personal-computer-tutor.com/abc4/v34/vic34.htm> "Disabling Error Reporting in Windows XP" by Vic Ferri. Go Start > Settings > Control Panel > System > W: System properties > T: Advanced > B: Error Reporting. Check 'Disable error reporting' and *uncheck* 'But notify me when critical errors occur'. Click [OK], W: System Properties, click [OK]. This prevents a 'wiimotelibpoll' error which is unimportant but annoying.

Mr. Ferri also suggests other ways of disabling error reporting and here is another good link: [http://www.windowsnetworking.com/articles\\_tutorials/Disable-Error-Reporting-Windows-XP-Server-2003.html](http://www.windowsnetworking.com/articles_tutorials/Disable-Error-Reporting-Windows-XP-Server-2003.html). Experiment with the settings until you have only disabled whiteboard stuff.

This is the content of the .reg file:

REGEDIT4

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\PCHealth\ErrorReporting]
"AllOrNone"=dword:00000001
"IncludeMicrosoftApps"=dword:00000001
"IncludeWindowsApps"=dword:00000001
"IncludeKernelFaults"=dword:00000001
"DoReport"=dword:00000000
"ShowUI"=dword:00000000
```

## 6.5 Wiiscantray

Wiiscantray is the ideal program for a busy teacher in a classroom. It permits her to just 'press the button' and, about 60 to 90 seconds later, to have her digital interactive whiteboard fully operational.

This is done by having a shortcut to wiiscantray.exe in the Start Menu or on the desktop. After powering up the PC, wiiscantray is automatically executed. It powers up the Wiimote and establishes the Bluetooth connection between the PC and the Wiimote.

When you have made the shortcut to Wiiscantray.exe, either on the desktop or in the Start Menu, and the USB power module and the Wiimote are started with wiiscantray.exe, you find the icon in the taskbar notification area *and* you find the pen software on the desktop.

## 6.6 Basic pen software

Finally, we should now examine the basic whiteboard software that steers the infra red pen. It's also started by wiiscan. The whiteboard software is C:\wiiscan\bin\Release\WiimoteWhiteboard\_v0.3.exe. The line that starts the whiteboard software is in wiiscan.ini and reads:

```
% and finally, a custom build v0.3 version, with fixed calibration.dat write
whiteboard_software=WiimoteWhiteboard_v0.3.exe
```

When you look around in wiiscan.ini, you see other whiteboard IR pen software versions, located in D:\. This gives you an indication how you can put your version of the IR pen software on the PC. This subject is further elaborated in section [13.1 IR pen software](#).

## 6.7 Problems and tips

When, for some reason, the complete set (either ready made or DIY) is not working, try to locate the error. Here are some tips:

- Read the excellent documentation in C:\wiiscan\doc. It was made for you.
- Enable debugging and logging:
  - In a DOS box, cd to C:\wiiscan\bin\release.
  - In wiiscan.ini enable debugging by setting <option\_debug=0 to option\_debug=1
  - In wiiscan.ini enable logging by setting option\_logfile= to option\_logfile=wiiscan\_log.txt
    - In C:\wiiscan\bin\Release\, you find the wiiscan\_log.txt file that may be of help.
  - Perform your actions, check the logfile and observe debug info.
- Open the wiiscan usage "help page" by typing in: <wiiscan -h or wiiscan -?
- Try the bluetooth scanning capabilities in the wiiscan software (scans for Bluetooth radios, and external BT devices).
- 

Another tip from Carsten

Many thanks for your feedback, let me see if I can help you. The logfile points me in the direction of either an undetected Bluetooth radio (adapter) or missing bluetooth link to the wiimote.

Try to scan for bluetooth radios using wiiscan.exe in a commando prompt:

```
> wiiscan -r
```

and then try to scan for external bluetooth devices like,

```
> wiiscan -s
```

The error

```
** error: BluetoothFindFirstDevice() returned null hbf
```

means that the software does not detect any bluetooth devices, the scan function ("wiiscan -s") should reveal any nearby devices and list them. You could try to put the wiimote into "discover" mode by pressing the "hard-sync" button (the internal red button) and then running "wiiscan -s".

You should then be able to see an wiimote adr. and perhaps the "Nintendo" name of the mote.

A short optimization would be to put the wiimote address (00:XX:YY etc.) into the wiiscan.ini file instead of

```
allowed_wiimote_adr = 00:00:00:00:00:00
```

This would allow wiiscan to find devices based on adr. and not on names only, and this mode is far more robust, when scanning for motes (the name part in the BT protocol takes a long time to transfer, and is in some cases missing when scanning....)

- A version of Boon Jin Goh's pen software always give a .NET Framework error when started via Wiiscantray and the path in wiiscan.ini. When pressing 'continue' in the error, BJ starts, but does not work properly. When starting BJ manually, no error.

#### SOLUTION 1:

There is a simple solution to most problems: create a batch file, for example c:\wiiscan.bat with the following content:

```
start /D C:\wiiscan\bin\Release wiiscantray.exe
wait -s 90
start /min /D C:\WiimoteSmoothBoard-0.2 WiimoteSmoothBoard-0.2.exe
```

NOTICE: We prefer the batch file below which gives even better results.

The first line starts wiiscantray.exe.

The second line gives wiiscan time to remove old Bluetooth connections and to start and create a new bluetooth connection. This is a secure way to create a connection. The time can be shorter; experiment with it.

#### SOLUTION 2:

Below is another, better way to start the whiteboard.

```
taskkill /f /im WiimoteSmoothBoard-0.2.exe /t
start /D C:\wiiscan\bin\release wiiscan.exe -wb -cf wiiscan.ini
C:\wait -s 30
start /min /D C:\WiimoteSmoothBoard-0.2 WiimoteSmoothBoard-0.2.exe
```

The first line kills the pen software. This is necessary in case of a second start of the batch file and prevents the existence of two or more instances of the Smoothboard program. For command line options type taskkill or see <http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/taskkill.msp?mfr=true>.

The second line starts wiiscan from the command line; that is without using wiiscantray. For command line options type start /? or see <http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/start.msp?mfr=true> .

The third line executes the wait program with a pause of 30 seconds. This pause depends on the PC used. For command line options type `wait -h` .  
And the fourth line starts the pen software..

Below is a batch file for a school that has many whiteboards, each with its own `wiiscan.ini` file. All programs are stored in the P: share on the server and in `SABRINE.BAT` the specific workstation name is stored.

```
taskkill /f /im WiimoteSmoothBoard-0.2.exe /t
CALL C:\SABRINE.BAT
start /D P:\ONSBORD\wiiscan\bin\release wiiscan.exe -wb -cf P:\ONSBORD\CONFIG\%
SABRINECOMPUTER%\wiiscan.ini
P:\ONSBORD\wait -s 30
start /min /D P:\ONSBORD\WiimoteSmoothBoard-0.2\WiimoteSmoothBoard-0.2.exe
```

#### NOTICE:

Sometimes even 90 seconds is too short. When Windows, or any other program starts looking for updates, or performs any other action, it is possible that Wiiscan is unable to start and the process is interrupted, resulting in an .

For this reason it is best to disable automatic updates by Windows and other programs. On [a href="http://support.microsoft.com/kb/306525"](http://support.microsoft.com/kb/306525) is described how:

Turn off Automatic Updates

If you want to turn off the automatic updates, follow these steps:

1. Click Start, click Run, type `sysdm.cpl`, and then press ENTER.
2. Click the Automatic Updates tab, and then click to select Turn off Automatic Updates.
3. Click OK.

The third line starts Boon Jin Goh version 0.2.

#### NOTICE:

Because Boon Jin Goh's software is started with the batch file, the software should not be started with `wiiscan` via `wiiscan.ini`. To eliminate starting any board software, open `wiiscan.ini` and search for:

```
% start wb at successfull connect, 0 or 1
        option_startwhiteboard=1
```

Set `option_startwhiteboard=1` to `option_startwhiteboard=0` (zero). This does not start any whiteboard software and prevents the .NET Framework error in BJJ software version 0.2.

When you put `wiiscan.bat` in the Start Menu, everything works by just switching the PC on and logging in. Of course you can also start some application with this batch file.

All programs, 'Wiiscan', 'wait' and 'WiimoteSmoothBoard' can be found on <http://wyxs.net/downloads/whiteboard/>. See also section [13.1. IR pen](#)

[\(top\)](#)

## 7. Infra red pen

The infra red pen replaces the mouse when using the whiteboard. Basically it consists of a LED (Light Emitting Diode), a 1,5 Volt battery and a momentary switch.

An abundance of infra red pens can be found on the Internet. Ready made, as well as do-it-yourself projects. A comprehensive how to on making an infra red pen for about € 4,-- can be found on [http://wyxs.net/web/wiimote/digital\\_whiteboard.html](http://wyxs.net/web/wiimote/digital_whiteboard.html).

In a year of experimenting, we have gained some experience with infra red pens. Here is our humble opinion:

1. The best infra red pen looks like a normal pen, feels like a normal pen and performs like a normal pen.
2. When lightly pressing the tip of the pen to the whiteboard surface, an internal momentary switch should put the LED on.

3. Manually operated switches make cheap pens, but are not easy to use.
4. A 1,5 Volt battery, type AA or AAA, is the best power source. It's available everywhere and cheap.
5. Use a VISHAY TSAL6400 high-power infra red LED. It has the right wavelength, lightning angle and power.

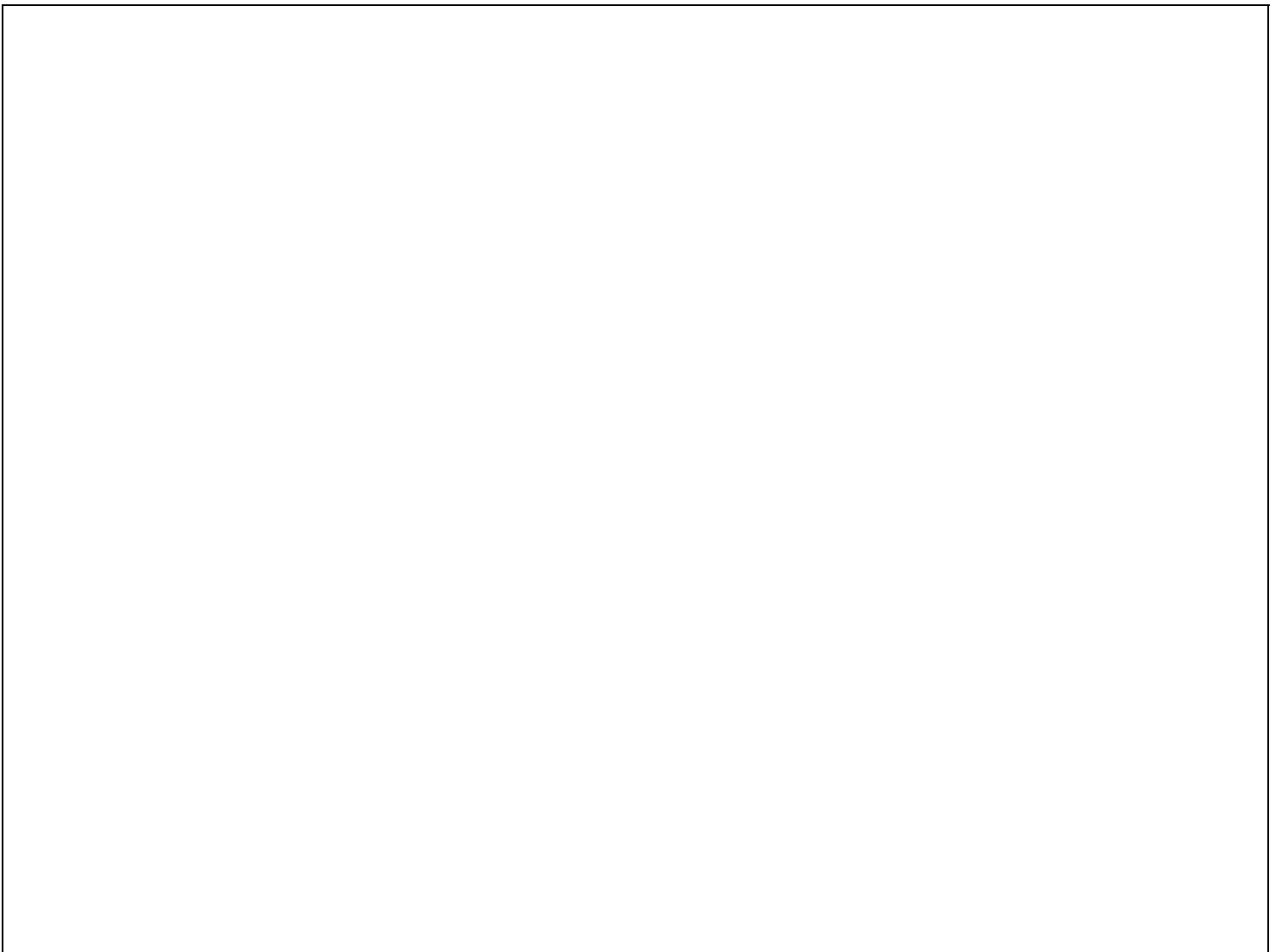
At this moment our school uses 'The Groove' from <http://www.irpenonline.com/>. A reasonable good pen. Not so good on the first point in the list.

[\(top\)](#)

## 8. Whiteboards

There are a lot of possibilities to get a whiteboard which is both a good projection surface *and* a good writing surface for an IR pen *and* being able to write with felt markers (both for whiteboards and 'normal' makrers which are sometimes accidentally used). We looked at reflective properties, ease of cleaning, cost effectiveness and last but not least the 'hot spot, which is exmplained at [http://en.wikipedia.org/wiki/Projection\\_screen](http://en.wikipedia.org/wiki/Projection_screen).

And finally, we wanted a whiteboard like this example:



blackboard.jpg

The blackboard at about 13.00 PM.





whiteboard\_ready.jpg

It took about an afternoon to make this whiteboard. The teachers like this one because they can still use the chalkboards left and right, fold them to use the backsides and have a perfect projection surface, that is easy to clean, even when someone uses the wrong felt marker.

Here is an overview of our experiences with whiteboard surfaces up to november 2011.

### 8.1 Melamine

We started our experiments with a chipwood board, covered with a Melamine layer, costing about € 11,-- (eleven) for a size of 244 x 122 centimeter. We just put it in front of the classical green blackboard. For testing purposes it's a perfect and cheap solution. **Advantages:** very cheap, reasonable reflective qualities and hot spot, can be sawn to any size.

**Disadvantages:** a bit difficult to clean. Do not use an abrasive (Jiff). A mild detergent will do. One can expect the Melamine layer to become thin over time, requiring replacement of the board. We do not use this solution any more.

### 8.2 (HPL) High Pressure Laminate

HPL, often misnamed 'Formica' (a brand name) has a very hard and smooth surface and is available in various shades of white, also dull to reduce the hot spot. It's about a millimeter thick and can be glued on the blackboard, see [Glueing HPL](#) below..

#### RESOPAL

A HPL from 'Resopal' (white-dull No. 0179-20), costing about € 15 to 20 per m2 including VAT. Available in 365 x 132, 305 x 132 and 218 x 102 centimeter, thickness 0,7 millimeter. No hot

spot, very good projection quality. Cleans well, see [8.2.1 Cleaning HPL](#) below. This is a very good surface.

#### DUROPAL

Duropol (No. W400 SM). Very good projection quality and no hot spot, cleans like a blackboard, see [8.2.1 Cleaning HPL](#) below.. Maximum size 410 x 130 centimeter, thickness 1 millimeter. Price per m2 including VAT around € 16. This is up till now (december 2011) the best non-magnetic surface.

#### HOMAPAL

We found Homapal (No. 8217). A magnetic HPL. The magnetic layer is an advantage of this brand, but it has a bit of a hot spot. Maximum size 305 x 130 centimeter, thickness 1,1 millimeter. Price inc. VAT per m2 around € 40. Cleaning, see [8.2.1 Cleaning HPL](#) below.

### 8.2.1 Cleaning HPL

HPL can be cleaned with hot water and common *non* abrasive household cleaners. The only precaution: avoid using strong acids and bases (adapted from: [http://www.arpausa.com/hpl\\_arpa/\\_pulizia.asp](http://www.arpausa.com/hpl_arpa/_pulizia.asp)).

When you accidentally have used a permanent felt marker, you can remove your writing with a whiteboard marker and hot water.

On <http://www.skovby.dk/showpage.php?lang=NL&pageid=7031526> is a Danish, English and German extensive PDF document on maintenance. Also available [here](#)

Duropol gives cleaning instructions on <http://duropol.b-onweb.de/care-cleaning>

### 8.2.2 Glueing HPL

We found Dutch illustrated guidelines on how to glue HPL on <http://www.westerhouthandel.nl/Links-menu/Zo-werkt-dat/Zo-plak-je-Hpl.html>. Please help us to find an English manual on glueing HPL (mail us).

## 8.3 Enamel

These boards are made of a layer of enamel on a sheet of metal. This is the 'whiteboard' as most of us know it.

**Advantages:** easy to clean, small magnets can be used to attach paper etc. to the board

**Disadvantages:** expensive, reasonable reflection, can have a big hot-spot, heavy.

We tested an example from [Smit Visual Supplies](#) who claims excellent reflective and hot spot properties. Reflective properties are good, but the dull enamel coating produces a rather big hot spot; bigger than those of HPL and the EGAN foil below.

## 8.4 Paint

Whiteboards based on a hardened paint on steel exists. We have no experience with this type of whiteboard. Please send us an example.

## 8.5 Plastic foil

We tested EGAN Visual Surface Technology's self-stick adhesive material. It can be used as whiteboard and projection surface and can be cut to the size of the blackboard you are using now. It may sound vulnerable, but it's not. Our Belgian colleague Kurt (<http://digitaleschool.be>) tested the surface with ballpoints, permanent felt markers and a knife. It proved *not* knife-resistant. In our experience, over time the surface deteriorates a bit due to pupils using ballpoints, et cetera.

**Advantages:** excellent reflection and low hot-spot, easy to clean, can be cut to any size (maximum length 32 meters(!) x 137 centimeters). **Disadvantages:** VERY expensive, 2 x 1 meter is about € 400 (four hundred euro), not that classroom resistant. Our first board produced a few tiny and almost invisible air-pockets. This was due to the material we attached the foil to. When using a chipwood or MDF board you get a flat surface.

#### OUR CONCLUSION:

Duropol and Resopal are our favourites.

[\(top\)](#)

## 9. Speakers

Simple speakers are good enough. Hi-Fi quality is not needed even not desirable. The low frequencies in the audio spectrum lessen the intelligibility of the sound.

### 9.1 Speaker in the projector

There are projectors with built-in speaker. That saves you work. A small disadvantage may be that the sound is not coming from the screen but from the ceiling of the classroom. With short throw projectors this disadvantage is minimised because the projector is near the screen.

We have now about a year experience now with the built in speaker in our short throw beamer and we absolutely prefer this way of working over separate speakers. The sound volume is controlled with the beamers remote control.

### 9.2 Separate speakers

You can use a speaker set that is normally used with the PC. Attach the speakers on top or next to the whiteboard. Fixing the speakers near the whiteboard improves the impression of the sound coming from the screen. Use extension cables to get the sound signal from the PC to the speakers.

A disadvantage can be that the volume control is near the whiteboard. One can also see it as an advantage; standing next to the whiteboard.

[\(top\)](#)

## 10. Ceiling mounts

There is an abundance of ceiling mounts. We go for simple and cheap. Most times the ceiling mount that comes with the video projector is quite expensive. The chosen mounts are universal and in use at our school.

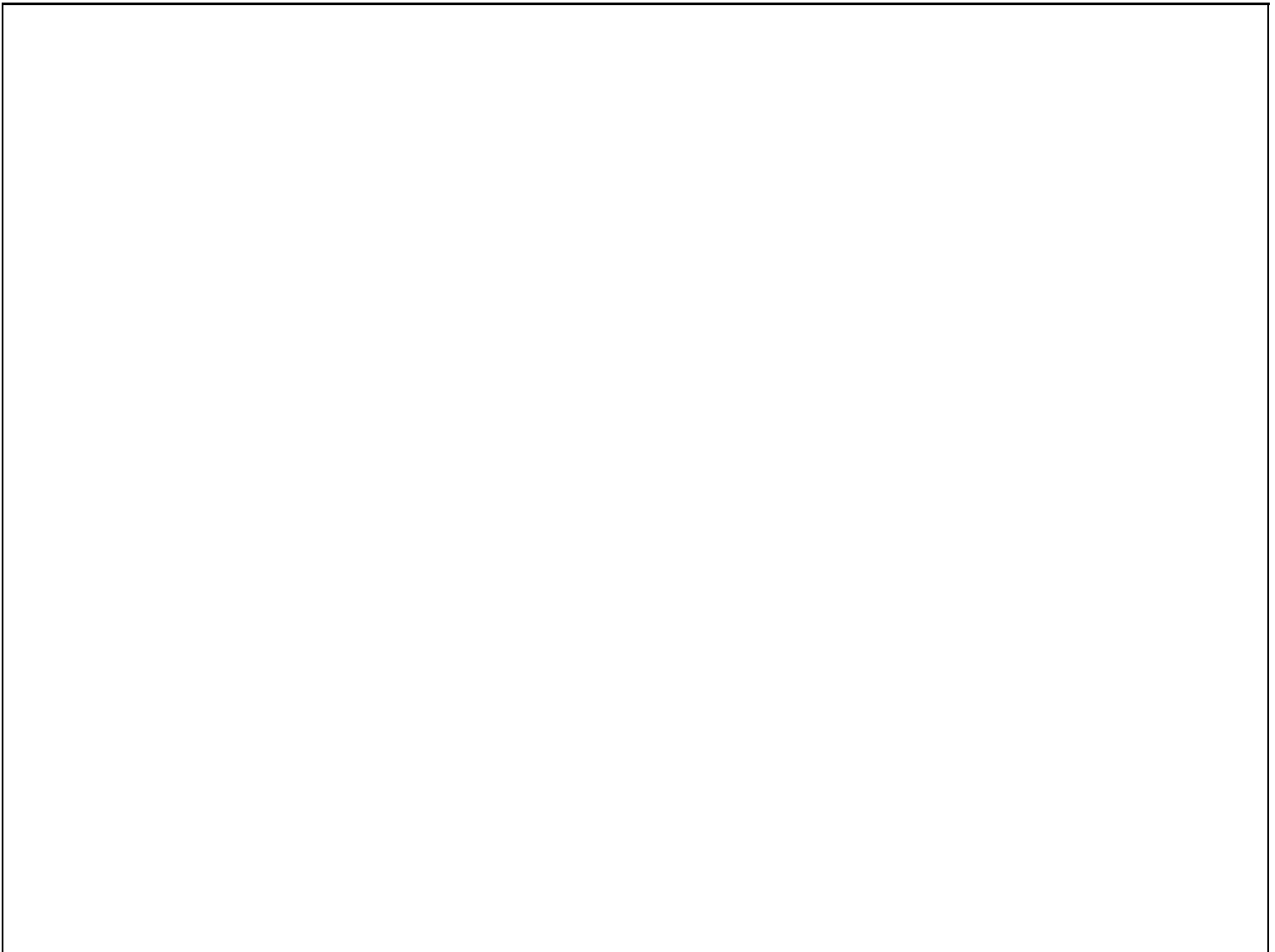
### 10.1 Wiimote ceiling mount

The wiimote weights virtually nothing, so any cheap ceiling mount will do. You can use the same mounts that are use for surveillance or Closed Circuit Tele Vision (CCTV) cameras. The mount must have a ball head for precise adjustment of the viewing area. Ceiling mounts cost between € 6 and € 10. Below is the easy mount we use, the SEC-BRACK50 (Google is your friend) :



wiimote\_ceiling\_mount\_SEC-BRACK50.jpg

It can be taken apart ...



wiimote\_ceiling\_mount\_SAC-BRACK50\_pieces.png

... and easily extended when the classroom has a high ceiling. Go to an ironmonger and buy an 'extended nut', diameter M8 (8 millimeter).



extended\_nut1.png

Also buy a meter of M8 thread bar (is that the correct name?). Total costs about € 3.

[ SAC-BRACK50 taken apart ]

extended\_thread.png

If you want, you can cover the thread with a chrome pipe.

## 10.2 Video projector ceiling mount

We found a simple, cheap, universal and sturdy video projector ceiling mount at <http://www.extreme-shop.nl/>. The type to order there is the BEAMER-P20.



They sell between € 30,-- and € 50,-- inc. VAT, depending on quantity.

## 11. Video projector

The most expensive part of your setup, so give it a lot of attention.

- Short throw beamers are better for teachers and pupils. Manufacturers of projectors, in their user guides, warn not to look into the projection lamp. That's clear language. In several countries it's forbidden for teachers and pupils to look into the video projector lamp.
- When you want an old fashioned projection setup, no whiteboard interaction, no pupils presenting in front of the whiteboard, then take an old fashioned projector with a long throw lens.
- The resolution should be at least 1024 x 768 pixels.
- Do not be impressed by the number of ANSI lumen, but test projection qualities in the classroom, with the whiteboard, in the expected lighting conditions, to see if the projector produces enough lumen for your purposes. Remember that how bigger the screen, the more light is needed.
- In a bright well lit classroom, is there some kind of darkening or dimming the light?
- Opt for a maintenance free projector. Some have filters that need regular cleaning. And, once every few months cleaning a projector, standing on a ladder, is a risky task for a teacher.
- How long is the expected life time of the lamp? And how much costs a new one? And can it easily be replaced by a teacher?
- Projectors can be noisy. Test for silence, or the lack of it.
- We like projectors with a VGA in- and output. In a classroom you also need a normal monitor and the return signal from the projector is used for the monitor on the PC. If your projector does not have a VGA output, buy a good quality VGA splitter. Cheap ones may produce low image quality.
- Built in speaker or separate speakers? Both have their dis- and advantages. If you use a long throw beamer, the sound of a built in speaker does not seem come from the screen.
- Aspect ratio. Do you want to continue using the old 4:3 aspect ratio or the new 16:9? Can the projector and the monitor handle the new aspect ratio? Testing might be necessary before drilling, cutting and sawing. Our blackboards measure about 2 x 1 meter which has some resemblance with the 16:9 aspect ratio.
- ...More tips? Please write.

We have good experiences with the OPTOMA EX525ST: short throw, maintenance free, built in speaker, VGA out.

[\(top\)](#)

## 12. Installing, connecting and testing

We have installed a couple of digital interactive whiteboards now. We have a bit of experience and want to share our successes, failures and errors in the hope you can do better. About one advice we are pretty convinced: test before you drill, cut and saw.

### 12.1 Whiteboard

No matter what kind of surface you choose, the place of the whiteboard in the classroom is important. The classical blackboard is installed on a place where it can get good daylight. In theory that's not the best place for a digital interactive whiteboard. Furthermore, the sun radiates a lot of infra red light; direct sunlight on the whiteboard might spoil correct pen tracking. Make a calulted guess about what the sun does during a year. Curtains or Luxaflex can help to dim the light a bit, but a classroom in the dark harms pupil/teacher interaction.

### 12.2 Video projector

Fixing a video projector to the ceiling is easy with a ceiling mount. The difficult part of the job is finding the right place. Do some real life experiments with the projector to find the right spot for the projector before drilling holes and cutting ceiling tiles. Be aware that the *lens* (and not the projector) should be exactly in the center of the projection area. A carpenter's square can be of service.

### 12.3 Wiimote

The buttons 1 and 2 need to be pressed permanently. One can use a piece of plastic and a tie-wrap or glue. Tape is easy but the risk is that, over time, the button pressure releases the tape. So, to your surprise, after hours of looking why your whiteboard is not working, you discover that the buttons are no longer pressed *enough*. Maybe it's a better idea to take a piece of white plastic and glue it over the buttons or tighten it with a tie-wrap. Do not glue the buttons themselves because glue may leak into the Wiimote.

A permanent solution is a piece of perspex as on this picture:



perspex\_wii\_permanent\_on.png

The wiimote can be attached to the ceiling via a mount, preferably with a ball head for easy positioning the Wiimote to the whiteboard. The middle of the desktop area, the lens of the projector and the Wiimote should be in one line if possible.

You have to test what is the best location for the Wiimote. Too near the whiteboard increases tracking resolution but you risk incomplete coverage of the desktop area and, if you use the 12 IR sensitive spots, the area around the desktop. Too far away results in low resolution. This is demonstrated in [http://wyxs.net/web/wiimote/digital\\_whiteboard.html#h6.1](http://wyxs.net/web/wiimote/digital_whiteboard.html#h6.1)

Fine tuning the position of the Wii takes some patience and can best be done with 2 persons; one behind the PC to start the calibration procedure and calibrating, the other person on a ladder to fine tune the position of the Wiimote.

For calibration you can best use the pen software that has a feature that shows the area covered by the IR camera in the Wii and the PC desktop area. The Smooth Board 0.2 BETA software has this feature.

### 12.3.1 Distance of Wii to whiteboard

What is the exact distance between the Wiimote and the whiteboard *in your classroom*? You can find it out by trial and error, or do some calculations to get an idea of the right spot. Anyhow, before drilling holes, test the found position on the ceiling with the Wiimote.

Thanks are due to work done by [Eyucel](#) and [Stefan van Aalst](#).

The IR camera in the Wii has an aspect ration of 4:3. The camera has a horizontal viewing angle of 40 degrees and a vertical viewing angle of 30 degrees.

NOTICE:

The caculations below assume the Wii is positoned under an angle of 90 degrees of the

whiteboard, on the crossing of the diagonal lines of the rectangle formed by the desktop and eventual sensitive areas around the desktop, i.e. a test position.



wiimote\_distance.png

The distance of the Wiimote to the screen can be calculated with the following formulas where:

d = distance

h = total horizontal screen size

v = total vertical screen size

For horizontal screen size:

$$d = \frac{h : 2}{40 : h}$$

And for the vertical screen size:

$$d = \frac{v : 2}{30 : v}$$

A classroom example: The projected horizontal width of the desktop image on the whiteboard is 130 cm. We also want a 10 cm area around the desktop as clickable areas. So, the total horizontal width becomes 130 + 10 + 10 = 150 cm.

$$d = \frac{h : 2}{40 : h} \quad d = \frac{150 : 2}{40 : 150} \quad d = \frac{75}{0,2666} \quad d = 281,32\text{cm.}$$

UNDER CONSTRUCTION!!

An example:

The screen size horizontal is 130 cm. The result of  $\tan(\text{radians}(20))=0.36397$

$$x = (130/2)/0.36397=178.50 \text{ cm.}$$

But, if the screen size vertical is also 130 cm, then  $x = ((\text{the screen size-vertical})/2)/\tan(\text{radians}(15))$

$$\tan(\text{radians}(15))=0.267949$$

$$x = (130/2)/0.267949=242.50 \text{ cm.}$$

These are approximate distances, but very close to the true numbers.



We now can calculate the distance of the Wiimote to a whiteboard when we know the total width of the projected desktop and clickable areas. However, bear in mind, the Wiimote is mounted to the ceiling, thus projecting downward. So, the IR camera sees a trapezoid like image of the desktop, where the bottom of the trapezoid is the widest.

The cable from the Wiimote to the USB power module can be led over the ceiling if possible. The cable is long enough to place the computer where you (or your pupils) need it. The cable can also be led to the video projector mount and led through its pipe.

#### 12.4 Speakers

- Where do we want the speakers? In the video projector, next to the whiteboard or ...?
- How do we do the volume control? A knob on the speaker, a separate amplifier with a volume control, with software on the PC?
- In a classroom one does not need high-fidelity sound quality; that even undermines intelligibility. Generally cutting off low frequencies improves intelligibility.

#### 12.5 Cables

You need quite a few rather long cables. Because they are long, the risks of signal loss and noise exist. Buy good quality audio and video cables.

- Where will the cables lay? In the classroom, via the ceiling, on top of the ceiling tiles, ...? It makes sense to find a way to get rid of the cables.
- How long will the cables be?
- It's not a bad idea to have a multi socket extension lead with a on/off switch to power everything with one switch. It saves energy.
- Double sided velcron tape is your friend when tidying up cables and giving your setup a final touch.

#### NOTICE:

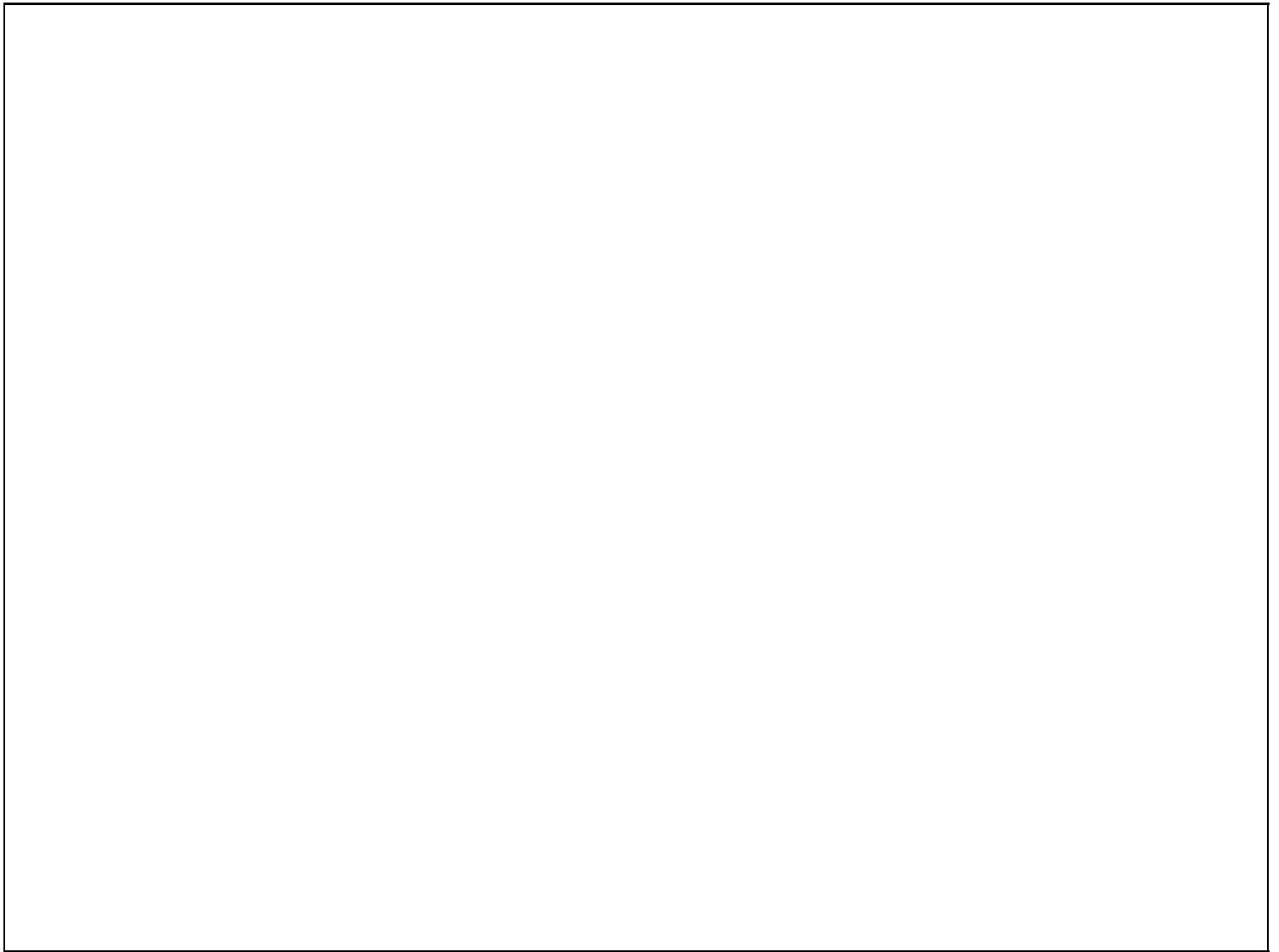
Observe local electricity regulations concerning earthing the projector, leading cables in a cable trough, and separation of mains and other cables.

#### 12.6 Testing

No need to say that extensive testing and a try out period are necessary before giving the infra red pen to the pupils and their teacher.

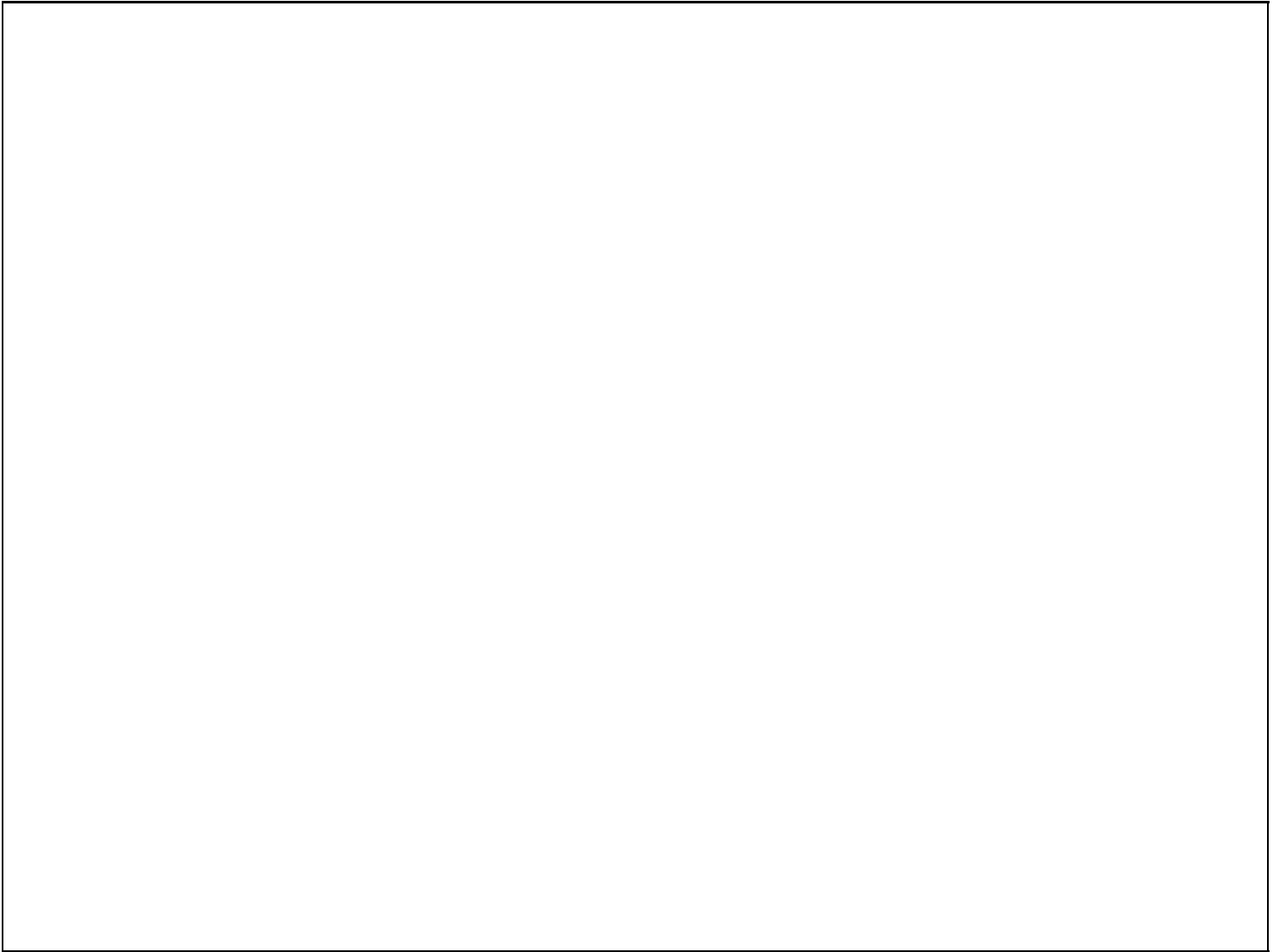
#### 12.7 Ready!

Here is our first setup with an old fashioned video projector with ONSBORD Wii setup <http://onsobrd.nl> and a formica whiteboard.



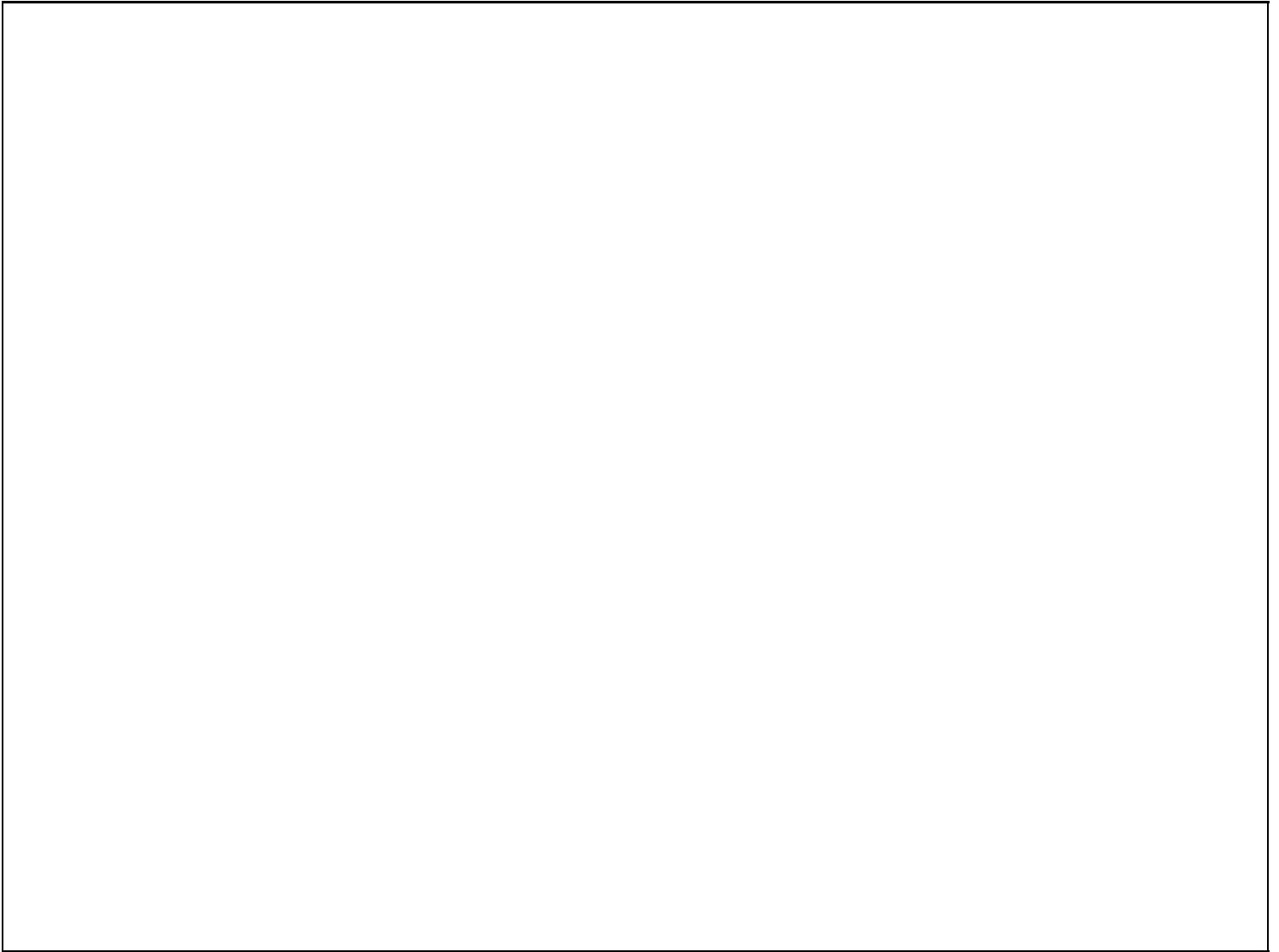
ONSBORD\_1\_ready\_front.jpg

The program to write with 'ONSBORD' (OURBOARD) is Jarnal, see <http://www.dklevine.com/general/software/tc1000/jarnal.htm>




ONSBORD\_1\_ready\_rear.jpg

The projector and Wii on the ceiling



short\_throw.png

Our other classroom with an Optoma EX525ST short throw projector with ONSBORD (OURBOARD)  
<http://onsobrd.nl>.



short\_throw\_high\_ceiling.png

A short throw projector on a high ceiling.

[\(top\)](#)

## 13. Software

### 13.1 IR pen software

We use Boon Jin Goh's Wiimote Smooth Board version 0.2 BETA, dated 13th July 2008. This version has enough features, right-click, double-click and 12 clickbale areas outside the desktop area are available. Our primary schools need no more.

You can download the 0.2 BETA version at <http://wyxs.net/downloads/whiteboard/>, where you find Wiimote\_Smooth\_Board\_0.2.zip 02-Jul-2009 06:36 267K .

NOTICE:

Use this version when you have tested everything with Boon Jin Goh's Smoothboard version which is included in Carsten Frigaard's Wiiscan software.

When you have everything working with the pen software included in Carsen's Wiiscan you can take the final steps: make a 'one click' solution by again modifying wiiscan.ini and creating a batch file. Proceed as follows:

Open Notepad and create a batch file, for example c:\startwb.bat. In it the following content:

```
start /D C:\wiiscan\bin\Release wiiscantray.exe
wait -s 90
start /min /D C:\WiimoteSmoothBoard-0.2 WiimoteSmoothBoard-0.2.exe
```

**Explanation:**

The first line starts wiiscantray.

The second line gives wiiscan time to start and create the BT connection. Ninety seconds is good for our setup; experiment with the time until you have no errors and a reasonable short start time. Be aware that Carsten chose the safest connection method, not the fastest.

The third line starts Wiimote Smooth Board version 0.2 BETA.

**NOTICE:**

Because BJG's software is started with the batch file, the software should not be started with wiiscan via wiiscan.ini. To eliminate the starting of board software with wiiscan.ini, open wiiscan.ini and search for the line:

```
% start wb at successfull connect, 0 or 1
    option_startwhiteboard=1
```

Set option\_startwhiteboard=1 to option\_startwhiteboard=0 (zero). This does not start the whiteboard software in Wiiscan and prevents the .NET Framework error in BJG software version 0.2 BETA.

To put this file in the Start Menu, right click on 'Start' > Open > Programs > Startup, and move the batch file there. The only thing a teacher has to do is to switch the PC on and log in. Of course you can also start some application with this batch file.

Both programs, 'wait' and 'WiimoteSmoothBoard' can be found on

Boon Jin Goh's Wiimote Smooth Board is now also available in a paid version as trialware with a nag screens. His latest version has lots of features, see <http://www.smoothboard.net/home/>.

**12.3 Mouse pointers**

An underexposed subject in digital interactive whiteboards. The pointer we normally use on the computer is created to be manipulated by the mouse. So there is nothing between your eyes and the pointer.

On a digital interactive whiteboard most times the pen is always where the mouse pointer is, thus the pen is hampering the view on the pointer. When you also are, like most people, right handed, you have difficulties viewing the pointer.

For these reasons you could use a pointer that is bigger, to improve visibility. When it's also pointing from a different angle (pointing towards the bottom right corner of the screen), we have a better view on the pointer.

Use 'Icon Edit' freeware and part of 'Icon Suite', created by Mike Bouffler. Google or see: [http://freewareapp.com/mb-icon-edit\\_download/](http://freewareapp.com/mb-icon-edit_download/). We use version 2.1.9 which enables us to create every pointer we want.

A few mouse pointers can be downloaded at <http://wyxs.net/downloads/whiteboard/>.

**13.3 Whiteboard software**

There is an abundance of whiteboard software; commercial, freeware and paid, open- and closed source, of differing quality and created for differing purposes. The list below is mainly for our purposes and reflects our views on software in education: preferably it should be open standards and open source.

- **Jarnal:** by David K. Levine and Gunnar Teege. From the blurb: "Jarnal is an open-source application for notetaking, sketching, keeping a journal, making a presentation, annotating a document - including pdf - or collaborating using a stylus, mouse or keyboard. It is similar to Microsoft Windows Journal and to the earlier Mimeo whiteboarding and Palm notepad applications." Site: <http://www.dklevine.com/general/software/tcl1000/jarnal.htm>

We love this fine cross-platform application. Dutch, German and Italian versions are available.

Below is a batch file that eases starting up Jarnal with options. and makes the command a more readablend. For markup reasons Below is a way to make the command more readable.

```
setlocal
set memory=mx192m
set jarnal=c:\jarnal\jarnal.jar
set template=c:\jarnal\templates\white.jaj %1 %2 %3 %4 %5
set background=c:\jarnal\backgrounds\graph7.jpg

java -X%memory% -jar %jarnal% -t %template% -b %background%
```

'Setlocal' enables local environments to be changed without affecting anything else.  
'Set' allows you to change one variable or string to another.

- **Whiteboard V1.0:** by Gary Jones. A kind of simple version of Jarnal. Good manual, Platform independent. Little features, which is a big feature. The ability to write outside the desktop area is a blockbuster in school.  
You can find Garys "Whiteboard Getting Started Guide", the "Windows Whiteboard Executable Launcher" and the "Whiteboard Application" on <http://www.wiimoteproject.com/index.php?action=downloads;cat=3>. Another download site is: <https://www.intellectfactory.com/>. It's a secure connection, accept Garys certificates.
- **Coccinella:** by Mats Bengtsson and now maintained by Sander . From the blurb: "The Coccinella project is [...] a free and open-source cross-platform communication tool with a built-in whiteboard for improved collaboration with other people." Site: <http://thecoccinella.org/>  
  
Try it and collaborate! Coccinella is available in many languages; also in Dutch.
- **GeoEnZo:** Excellent Dutch whiteboard software by Ton Groeneveld. From the blurb: " GeoEnZo is een programma waarmee het digitale schoolbord in de wiskundeklas als dagelijks instrument kan worden gebruikt. Met GeoEnZo heeft u het 'gewone' (white)bord eigenlijk niet meer nodig. Ook uw oude bordpasser, liniaal of geodriehoek kunt u voorgoed opbergen. GeoEnZo voorziet in alles wat u op het bord nodig hebt en biedt u daarbij heel veel elektronische voordelen." Software is worth translating.  
<http://geoenzo.nl/geoenzo/index.htm>
- **LinktivityPresenter:** From the blurb: "LinktivityPresenter is a personal graphic communications tool for everyone. You can draw, highlight and type directly on your computer screen, on PowerPoint slides, on Word documents, Excel sheets or on any application that you have open on your computer."  
Also from the page below: "Note: LinktivityPresenter was initially sold for \$9.99. Effective April 2006, it is now free."  
Site: <http://www.linktivity.com/presenter.html>.
- **Kindlelab:** says their whiteboard software is free and open source. I don't know if it is that free when it's not GPL licensed but has a 'Ning' license. You must enter a 'social network' to download it. Site: <http://kindlelab.com/>
- **Edusim:** thier blurb: "Edusim is a free opensource 3D multi-user virtual world specifically for your classroom interactive whiteboard. Edusim is extendable allowing multiple classrooms to connect their interactive whiteboards for collaborative learning session". The software is in alpha stage. Site: <http://edusim3d.com/alpha/>><http://edusim3d.com/alpha/>  
See also their download site at: <http://www.classroom20.com/profiles/blog/show?id=649749%3ABlogPost%3A116931>
- **Moodle:** OSS/GPL E-learning system is world famous. Better than Blackboard. Go Site: <http://moodle.org/>. Log in as guest and go to: <http://moodle.org/mod/forum/discuss.php?d=37248> to follow a discussion on whiteboard software. You can dowload whiteboard.swf, a Flash drawing program.
- **Eduribbon:** from the blurb: "EduRibbon is a universal tool for creating interactive presentations and explanation support". They have a free version: <http://www.dosli.eu/eng/products/eduribbon>. Seems usable feeware.
- : ..... RANDOM STUFF, MUST BE EDITED!

The Wiimote Forum <http://www.wiimoteproject.com/> offer valuable information, ideas, solutions for problems and help. On this site at Wiimote Project » Forum » Wiimote Projects » Wiimote Interactive Whiteboard » Wii remote digital whiteboard with IR LED pen for schools HOWTO can be found.

Loose ends:

Magic Whiteboard <http://www.magicwhiteboard.co.uk/>  
Static whiteboard foil

SubEthaEdit <http://www.codingmonkeys.de/subethaedit/>  
Kin of collaboration tool. Closed, Costly.

Jabber clients <http://xmpp.org/software/clients.shtml>  
Good list of Jabber clients. Coccinella is just one.

Zengobi Curio, <http://www.zengobi.com/products/curio/>  
Unclear site, unclear pricing, unclear ..

General Electric: imagination at work <http://www.imaginationcubed.com/>  
Online drawing, looks good. Where to download?

MSN Messengeers whiteboard, instructions on: [http://www.computeruser.com/todaystip/tip\\_view/f27dd29ee797453e3a78bdaa285bbca8](http://www.computeruser.com/todaystip/tip_view/f27dd29ee797453e3a78bdaa285bbca8) says:

MSN Messenger has a built in Whiteboard has a useful and inactive feature that allows you to share pictures and drawings in real time on an MS-paint like application window. MSN Messenger must be downloaded and installed to utilize any features. To use the whiteboard feature: \* Open the Actions menu and choose Start Whiteboard \* You will send a request to the other party, and if accepted, the shared session window will appear. Choose Whiteboard to begin the whiteboard session Download: <http://messenger.msn.com/download/getstarted.aspx>

Notatelt: <http://www.notateit.com/>  
Commercial licenses from € 25 to € 65 per license.

PhotoNote: <http://www.softpedia.com/get/Multimedia/Graphic/Graphic-Editors/PhotoNote.shtml>  
Process images of your whiteboard? Freeware \$ 50

Whiteboard Reviewer Edition: <http://wareseeker.com/Business-Finance/requirements-whiteboard-reviewer-edition-1.0.0.zip/326690>

"Requirements Whiteboard is a system for structured requirements management, authoring, review, documentation, discussion and publication. The Reviewer Edition can be used standalone, but is intended for use alongside the comprehensive Author Edition". Freeware.

Firedoodle: <https://addons.mozilla.org/nl/firefox/addon/4651>  
Firefox add on turns any website into a whiteboard. To save you need to register to a server.

ClearBoard: <http://www.softtouchit.com/xpe/portal/674eef30-1119-1000-8121-46648aa451c5>  
commercial, € 77, needs a photo camera, sort of note taking software?

VirtualBoard 1.11, <http://lightools.fredisland.net/index2.htm>  
"VirtualBoard was at the beginning a tool created for professional needs, to teach people how to use Windows, with projection on large screen: instead of using big explanations in an incomprehensible technical jargon for a newbie (example: " Click on 'Start' Menu on the left side of the taskbar, at the bottom of Windows desktop " ), it was more energizing and motivating to SHOW what must be done ( " look at the red arrow I've drawn on the screen, and click at the place the arrow indicates " ) ! All that with one imperative : minium of mouse clicks".

Bluechillies :<http://www.bluechillies.com/list/whiteboard.html>  
A collection of whiteboard stuff. Freeware, shareware.

FaceTop: <http://www.wired.com/science/discoveries/news/2004/07/64129>  
"...a new system that cleverly blends a video-conference feed with a transparent image of a computer desktop into one full-screen window".

Transparent video overlay over desktop: <http://www.codeproject.com/KB/audio-video/TransparentDesktopVideo.aspx>  
Article and code as follow up on FaceTop.

CapTrue: <http://www.batchconverter.com/CapTrue-download-73132.shtml>  
Screenshot capture, also video.



Desktop Pen: [http://www.download3000.com/download\\_46770.html](http://www.download3000.com/download_46770.html)

"Desktop Pen is a free tools that allows you draw directly on desktop. You may mark or highlight what you see on desktop immediately with different colors".

Annotate Pro V2: <http://www.annotatepro.com/>

A kind of Jarnal, commercial, expensive.

Ritepen3.0: <http://www.ritescript.com/Products/ritePen.aspx>

"Write anywhere on your screen in continuous words and phrases, in smooth electronic ink. Watch your handwriting converted into text and entered into any application". Kind of Jarnal, commercial.

Scientific Calc: [http://my.hrw.com/math06\\_07/nsmedia/tools/Sci\\_Calculator/Sci\\_Calculator.html](http://my.hrw.com/math06_07/nsmedia/tools/Sci_Calculator/Sci_Calculator.html)

Exactly what it is.

Cochinnella documentation: <http://www.linux.com/articles/58326>

Modest.

v2Whitebaord: <http://www.version2software.com/v2whiteboard.html>

"...sketch out how your computer network looks ... instead of trying to describe it. Or, what if you wanted to collaborate with a co-worker on the UI design of your next great piece of software".

<http://lighttools.fredisland.net/index2.htm>

Looks great!

### **Links van Matthijs:**

KlasCement.net <http://www.klascement.net/>

Crickweb | Flash Page <http://www.crickweb.co.uk/assets/resources/flash.php?&file=Toolkit%20index2a>

Digibord :: digibord.Yurls.net - Your Favorite Websites <http://digibord.yurls.net/>

Leidster -- Leidsterstartpagina.jouwpagina.nl <http://leidsterstartpagina.jouwpagina.nl/>

Digitaal schoolbordonderwijs: Leermiddelen <http://www.digitaalschoolbordonderwijs.nl/materialen/>

Jouw Wereld Mijn Wereld <http://www.jouwwereldmijnwereld.nl/>

Digitaal schoolbord? Digibord op School <http://www.digibordopschool.nl/default.asp?target=1>

- - - ALLES OVER VERKEERSBORDEN - - <http://proto.thinkquest.nl/%7Eklb019/>

JigZone.com Jigsaw Puzzle Games <http://www.jigzone.com/>

Welcome to MirandaNet <http://www.mirandanet.ac.uk/index.htm>

Kent NGFL ICT Website <http://www.kented.org.uk/ngfl/index.htm>

kleuters <http://www.nldata.nl/laurentius/piramide/cgi-oic/pagedb.exe/show?no=627&fromno=506>

SMART Board Podcasts ? RedbridgePrimaryICT <http://redbridgepict.wordpress.com/category/smart-board-podcasts/>

PO 34 | Groep 3-4 in de praktijk | Wereldori?ntatie [http://digischool.kennisnet.nl/community\\_po34/indepraktijk/pagina](http://digischool.kennisnet.nl/community_po34/indepraktijk/pagina)

END LINKS MATTHIJS

[\(top\)](#)

## **14. Advanced topics**

### **14.1 Preventing RF emission**

Bluetooth operates with high- and low band RF signals, thus emitting RF energy. We have some ideas on omitting RF signals. These might be interesting for those who want to use our digital interactive whiteboard in environments like radio astronomy, or where regulations are against wireless communication. When interested, contact us.

#### 14.2 Working with the PixArt camera directly

Johnny Chung Lee provides an interesting but complex solution on

[\(top\)](#)

### 15. Testimonials

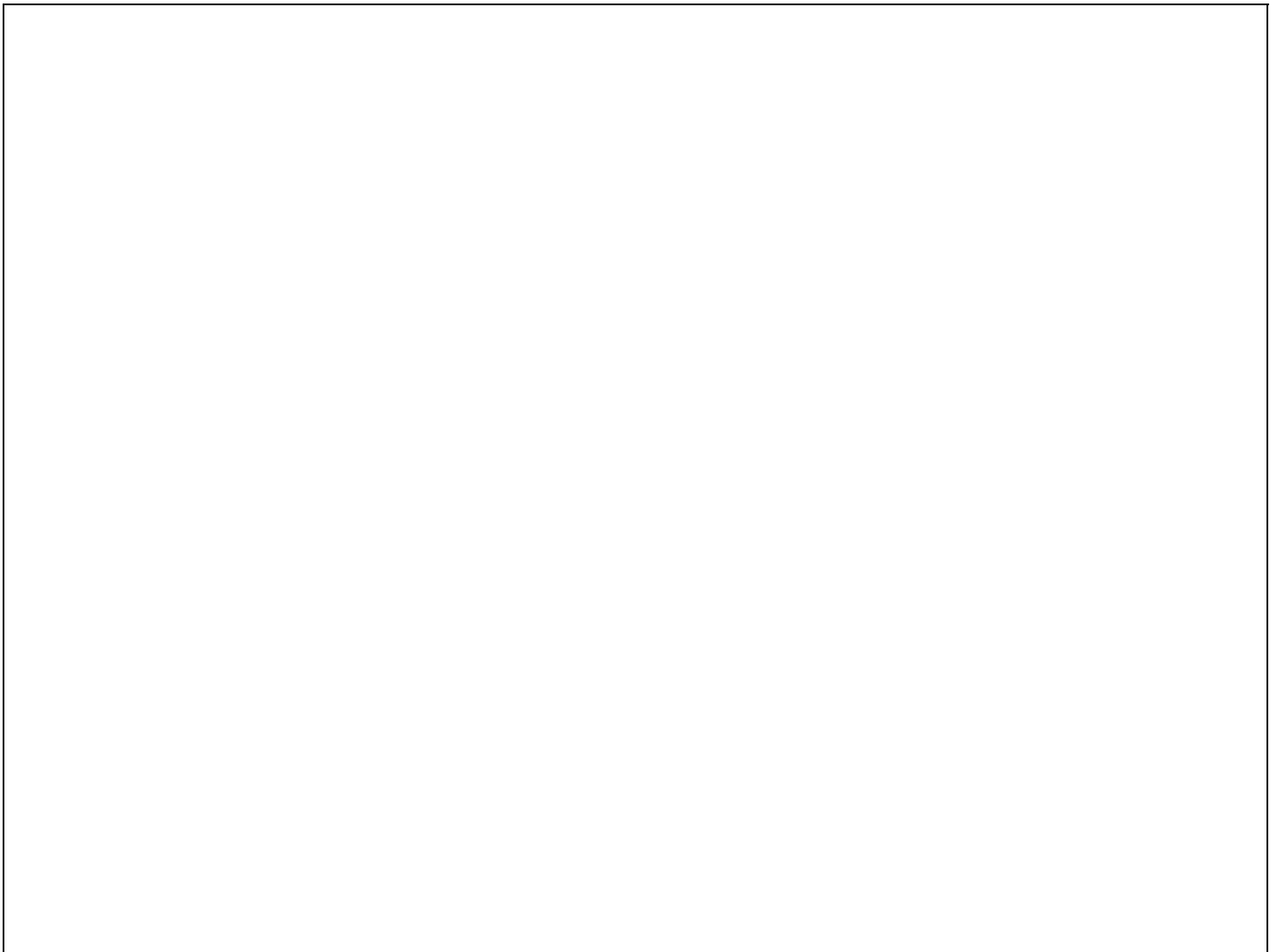
We received an e-mail from Aaron Johnson from the [Jordan catholic school](#) USA, from which we quote:

Anyways, it's awesome to hear that you're excited to see what we've done with your ONSBORD system in our school! We have your system set up in a 7th/8th grade classroom at our Middle school and it is the epitome of interactive whiteboards in our opinion! We plan to put the 3 additional systems in three more of our Middle school classrooms and will eventually bring the system to our 4th, 5th, and 6th grades.

[snip]

We really appreciate everything you have done to help improve free and open source technologies in education!

Aaron C. Johnson, Catholic School



[Jordan\\_school\\_classroom.png](#)

Aaron also sent us this picture of his classroom. His whiteboard has a bit of a hot spot.

We recommend visiting Aarons Technoly Center at <http://www.jordanschool.com/technology-center/wiiboard-resources/>. Many useful links and a quote:

"ONSBORD - The ultimate WiiBoard implementation! ONSBORD is what Jordan Catholic School is using in their classrooms to create a high-tech learning environment also known as the Interactive Whiteboard - we like to call it the WiiBoard! :~) ONSBORD has taken the original Wiimote Interactive Whiteboard idea (founded by Johnny Chung Lee) and has perfected it by making it work seamlessly in the modern day classroom. Once installed properly ONSBORD can and will function better than the commercial products provided by SMART and even Promethean! Please note that their main website is in Dutch so be sure to copy and paste the URL into Google Translate in order to be able to read it! Be sure to check out their DIY guide located here."

[\(top\)](#)

## 16. Concluding remarks

We have tried to put all we know about digital interactive whiteboards using a Wii in this how to. Comments and suggestions are welcome. Please write to ict [at] rosaboekdrukker [dot] net or schoutdi [at] knoware [dot] nl.

[\(top\)](#)

## 17. Disclaimer

Although we have tested every part of this project and many eyeballs have seen the texts, neither the author(s), nor the Public Primary School Rosa Boekdrukker or other schools or companies mentioned in this DIY project can be held responsible for any damage, neither direct, nor indirect, neither consequential, nor accidental or whatsoever, when using the hardware, software, materials, advices or suggestions we write about or refer to.

This project is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

[\(top\)](#)

Copyright © 2009, 2010, 2011, 2012 Dirk Schouten



This work is licensed under a [Creative Commons Attribution-Noncommercial-Share Alike 3.0 Netherlands License](#).

[\(top\)](#)

---

*Author: Dirk Schouten <schoutdi at knoware dot nl>*  
Last updated: 2011-11-27