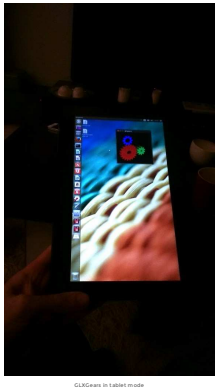
JOHN WELLS
KNOWLEDGE MANAGEMENT AND SOCIAL WEB APP GURU

WP-UNITED

UBUNTU (OR OTHER LINUX) ON THE ASUS TRANSFORMER BOOK T100

The T100 is a nice little convertible tablet/netbook.

My aim is to get a "perfect" Ubuntu installation on the T100, such that it can be used successfully as a daily machine in both netbook and tablet modes.



GLXGears in tablet mode

Unfortunately Linux support right now is quite rudimentary, and installing it requires us to jump through a few hoops. The situation is improving rapidly however. So I will keep updating this How-to as new drivers become available and things improve.

Warning

Eventually I expect hardware support for the T100 will be excellent, but we are not there yet. In order to get the best support possible, we will be using bleeding-edge builds and the latest Linux kernels. If you'd just prefer an easy life, come back in October and just install Ubuntu 14.10.

That said, this little convertible is a lovely machine, and Ubuntu/unity works very nicely on it – finally Unity has a purpose! The more people get on for the ride now, the quicker we can test and iron out bugs.

***** This post will constantly be updated as in-kernel support improves *****

Current status (updated 23/03/2014)

I'll update this whenever I manage to get new things working. I will only add items to the How-to below that are confirmed to work well.

- **Graphics**: Working with accelerated (3D) graphics **7/10**
- **Wifi**: Working, but often drops connection [working on improving this] **6/10**
- **Touchscreen**: Working, with multi-touch out of the box **10/10**
- **Sound**: Working, with patches **8/10**
- **SD card reader**: Working, some configuration needed (thanks akira) **9/10**
- **Battery monitoring**: Working, with patches **8/10**
- **Tablet keys (Volume up/down etc)**: Not yet working **0/10**
- **Power management (Suspend/resume)**: Not yet working **0/10**
- **Orientation sensor**: Not yet working [currently testing] **0/10**
- **Backlight & ambient light sensor**: Backlight not adjustable. Light sensor works with custom driver **5/10**
- **Touchpad**: Working, no multitouch yet **8/10**
- **Shutdown / reboot**: Working, with patches **9/10**

1. First steps: Preparing for the Ubuntu

Install

First things first, update using Asus LiveUpdate to the latest "BIOS" available. At the time of writing, that is v304. Do any backing up of Windows / recovery partitions. I'll leave the details of that up to you.

Before we attempt to boot Linux on the T100, we need to do some preparation, so start in Windows. Download the latest daily AMD64 build of Ubuntu 14.04 from here.

Download the Rufus USB bootable image creator, and "burn" your downloaded ISO to a spare USB stick. In Rufus, for "Partition scheme and target system type", choose "GPT partition scheme for UEFI computer". For "File System", choose "FAT32", and leave the rest at default. At the bottom, check "Create a bootable disk using: ISO Image" and select your downloaded Ubuntu image, then hit "Start".

When your USB stick is ready, close Rufus. It should now be browseable in Windows. Browse to the EFI\Boot directory, and place this bootloader (named bootia32.efi) there. This bootloader was compiled from source using the latest Grub2. If you don't trust random downloaded files from the Internet (and you shouldn't!), you can find the instructions for building it yourself here.

2. Booting the Live Image

Now, insert the USB stick and reboot to the firmware (BIOS). You can do this in Windows by holding shift when pressing "restart", then touching Troubleshoot → Advanced Options → UEFI Firmware Settings → Restart.

Once there, disable SecureBoot, then visit the boot options, and ensure the USB stick is the first in the list.

Press F10 to save settings, and after a few seconds you will be in the GRUB bootloader. Before the timeout, immediately hit CTRL-ALT-DEL. This will reboot the computer again, but this time you will have the laptop's native resolution (rather than being stuck at 800×600 from the "bios").

In the GRUB menu, highlight "Try Ubuntu", and press "e" to edit it. In the editing screen, scroll down to the command line options, where it says "quiet splash". Delete "splash" and replace it with:

```
video=VGA-1:1368x768e reboot=pci,force
```

Then press F10 to boot. You should get all the way to the Desktop.

3. Installing the distro

Click the "Install Ubuntu" desktop icon to install Ubuntu permanently.

The partitioning scheme you choose is up to you – but you will need to preserve the EFI partition, so don't just partition the entire disk for Ubuntu.

In addition to the EFI partition, I prefer separate /, /home and /boot mount points; but that is up to you. You could squish down the Windows partition and created the additional partition(s), or just delete the Windows partition altogether if you don't need it.

When done, reboot, leaving the USB stick in.

4. First boot

Ubuntu won't boot yet. We'll need to compile our own bootia32.efi to use with Grub. To do that we really need a wireless connection. So we'll boot manually, fix up wireless, and fix Grub.

Boot back to the Grub welcome screen on the USB stick. Hit `c` to drop to a Grub command line.

You'll need to provide Grub with the path to your kernel and initrd to boot. First, the path to the kernel:

```
linux (hd2,gpt5)/boot/vmlinuz-3.13-xxxx root=/dev
/mmcblk0p5 video=VGA-1:1368x768e reboot=pci,force
```

Here, (hd2, gpt5) refers to the fifth partition on the third disk (Partition numbering begins at 1 and disk numbering begins at 0). This will vary depending on how yo uinstalled and your T100 model. On my 32GB model, Grub assigns the USB stick as hd0, the read-only recovery flash chip as hd1, and the main internal flash as hd2. gpt5 is the fifth partition, but it will depend on how you installed.

Fortunately, grub has good auto-completion features, so you can hit twice as you type, and grub will list possible completions for you – just keep trying until you see the various vmlinuz kernels.

The root=/dev/mmcblk0p5 will also depend on the partition you installed to. It will be your root partition. Unfortunately this can't be auto-completed, so if you

can't remember your partition setup, you'll need to try by trial and error.

To complete the line, press Enter.

Then you need to specify the location of your initrd. This is easy, it's in the same place as the kernel:

```
initrd (hd2,gpt5)/boot/initrd-3.13-xxxx
```

Then Enter.

Then boot with:

```
boot
```

With luck after hitting Enter, you'll boot through to Ubuntu. If not, don't be disheartened – keep trying.

5. Enabling wifi

To get further, we'll need wifi. However internal wifi on the T100 isn't terribly reliable under Linux yet. If you have access to another wifi dongle, you might want to try that until you're fully set up.

The driver is already included, but it needs some firmware and a copy of your system's wifi nvram:

* Grab the Wifi firmware here (from the Linux-wireless repository). Copy it to /lib/firmware /brcm/brcmfmac43241b4-sdio.bin
* The NVRAM is difficult to get right now, as 32-bit EFI runtime services aren't yet available on 64-bit. But you can use my NVRAM – get it here and copy it to /lib/firmware/brcm/brcmfmac43241b4-sdio.txt

You'll now need to reboot using the same procedure again, and wifi should be working.

However, I found wifi performance to be very poor – there are some bugs in the driver. This is improved somewhat with the latest kernel, which we will upgrade to in a later step. I managed to improve performance somewhat by using the NVRAMs from other platforms, and splicing in some of the missing variables. You can see my tries here. In all cases, copy them to /lib/firmware/brcm/brcmfmac43241b4-sdio.txt , and reboot or unload/reload the wifi driver:

```
sudo modprobe -r brcmfmac
sudo modprobe brcmfmac
```

If you find a firmware/nvram combination that works really well, please share.

6. Completing the installation

We can't keep rebooting like this... so let's fix Grub. This assumes you now have a working internet connection.

We'll need some build tools – install them:

```
sudo apt-get update && sudo apt-get install git bison
libopts25 libselinux1-dev autogen m4 autoconf
help2man libopts25-dev flex libfont-freetype-perl
automake autotools-dev libfreetype6-dev texinfo
ia32-libs build_essential
```

Then get the Grub source:

```
git clone git://git.savannah.gnu.org/grub.git
```

Now build it:

```
cd grub
./autogen.sh
./configure --with-platform=efi --target=i386
--program-prefix=""
make
```

And install to efi:

```
cd grub-core
sudo ../grub-install -d . --efi-directory /boot/efi/
--target=i386
```

This will create a directory, 'grub', in your EFI partition. We want to copy the grubia32.efi from there to the location Ubuntu created during installation:

```
cd /boot/efi/EFI
sudo cp grub/grubia32.efi ubuntu/grubx64.efi
```

This should be enough to allow you to boot from the "ubuntu" option in your EFI firmware.

Before you boot, let's add the default command line options to Grub.

Open /etc/default grub in a text editor:

```
sudo nano /etc/default/grub
```

And edit the GRUB_CMDLINE_LINUX_DEFAULT exactly as we did before. When done, hit ctrl-o to save then ctrl-x to exit. Then, to update Grub:

```
sudo update-grub
```

Congratulations! you should now be able to boot/reboot directly to the Ubuntu desktop!

7. Upgrading to the bleeding edge to improve hardware support

There are still a lot of things to get working. The best way to improve hardware support further is to use the latest development branches of the kernel.

Unfortunately (as at the time of writing), even the latest development Linux kernel (3.14-rc5) lacks some key hardware support for Baytrail tablets. But we can improve that by pulling in latest patches for sound. We'll also compile in a new experimental feature for accessing 32-bit EFI services from a 64-bit system, as we need that for tools like efibootmgr to work. Finally, we'll pull in the latest wireless fixes and the latest power management changes. We need to power management changes for our battery patch.

I recommend you compile this on a fast desktop computer with a decent network connection. You could do this on your tablet, but it would be extremely slow. For the following instructions, I assume you're using a recent 64-bit Ubuntu on a desktop. If you're using 32-bit, you'll have to change the instructions to cross-compile – I'll leave that up to you (and Google).

On your compiling machine, install the packages you'll need to build a kernel:

```
sudo apt-get install git build-essential fakeroot
crash kexec-tools makedumpfile kernel-wedge
libncurses5 libncurses5-dev
sudo apt-get build-dep linux-image-$(uname -r)
```

Then create a working directory, step into it, and download the sources you'll need:

```
mkdir kernel
cd kernel
git clone git://git.kernel.org/pub/scm/linux/kernel
/git/torvalds/linux.git
cd source
git remote add sound git://git.kernel.org/pub/scm
/linux/kernel/git/broonie/sound.git
git remote add efi git://git.kernel.org/pub/scm/linux
/kernel/git/mfleming/efi.git
git remote add pmfixes git://git.kernel.org/pub/scm
/linux/kernel/git/rafael/linux-pm.git
git remote add wifinext git://git.kernel.org/pub/scm
/linux/kernel/git/linville/wireless-next.git
git fetch sound
git fetch efi
git fetch pmfixes
git fetch wifinext
```

Then merge the updates from sound, efi mixed-mode, power management and wireless onto the mainline kernel:

```
git merge sound/topic/intel
git merge efi/efi-for-mingo
git merge pmfixes/linux-next
git merge wifinext/master
```

There are still some patches we need to apply. Step back and create a patch directory, and download some patches into it:

```
cd ..
mkdir patches
*mkdir^cd
git clone https://github.com/jfwells/linux-
asus-t100ta.git
```

Now apply the patches:

```
cd ../source
git apply ../patches/linux-asus-t100ta/patches/*
```

They should all apply without error.

Now we're almost ready to compile. Before we do, we need a .config file to tell the kernel build system what parts we want to compile. Start by copying over the current one from your T100. You'll find it at /boot/config-xxxxxx , where xxxxxx is the version of the currently running kernel. Grab it and copy it to the kernel/source directory you've just been working in. Rename it to .config .

Now we need to update the config.

```
make oldconfig
```

This will prompt you to set configuration for the things that will be newer in this kernel. Select 'm' to compile relevant drivers as a module where offered (in particular all the 'soc' sound options). Select 'Y' to enable EFI mixed mode where offered. Leave the kernel debugging options off.

When done, we're ready to compile. We can just do it the quick and dirty way:

```
make clean
make -j8 deb-pkg
```

The above assumes this is on a quad-core processor with hyperthreading (hence the "-j8", or 8 simultaneous processes). Reduce this number appropriately if you have less cores.

This will take some time (anywhere from 20 minutes to over an hour. If you're doing it directly on the T100, all bets are off – it could take all day.

When complete, you'll have a set of .deb packages in the directory above (i.e. in the "kernel" directory). We need "linux-headers-xxxxx.deb" and "linux-image-xxxxx.deb". Choose the smaller of the linux-image packages, the one without debug symbols. Copy them over to your T100.

On the T100, install the new kernel with:

```
cd <folder where you saved the .debs>
dpkg -i ./*.deb
```

Then reboot!

8. Sound

Sound won't yet be working. First, you'll need some more firmware. For now, you can grab it from ChromiumOS, here. Unpack the archive and copy the files to /lib/firmware/intel .

The drivers are in a bit of a raw state – they expose about a hundred oddly-named mixer and DSP devices to ALSA. We can set some defaults. Grab the defaults file here, and copy it to /var/lib/alsa/asound.state .

Reboot, and force-load the state file into alsa with :
`sudo alsactl -f /var/lib/alsa/asound.state restore`

For me, my sound still sounds pretty awful. I need to mute the distorted right channel in alsamixer, then at least it is bearable. I also find the CPU runs a bit hot with sound enabled. Let me know if you have more luck.

*** please be careful – my sound is permanently distorted, even in Windows now; keep the volume low when testing the various options ***

9. SD card reader

The SD card reader works with newer kernels, but the sdhci module needs some configuration. Create a new file:

```
sudo nano /etc/modprobe.d/sdhci.conf
```

Then add the following:

```
options sdhci debug_quirks=0x8000
```

ctrl-o then ctrl-x to save and exit, then:

```
update-initramfs -u -k all
```

Then reboot and the card reader should be working.

9. Other drivers

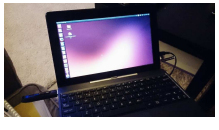
So far, the only additional driver I have working is the light sensor. Please download and test my driver here.

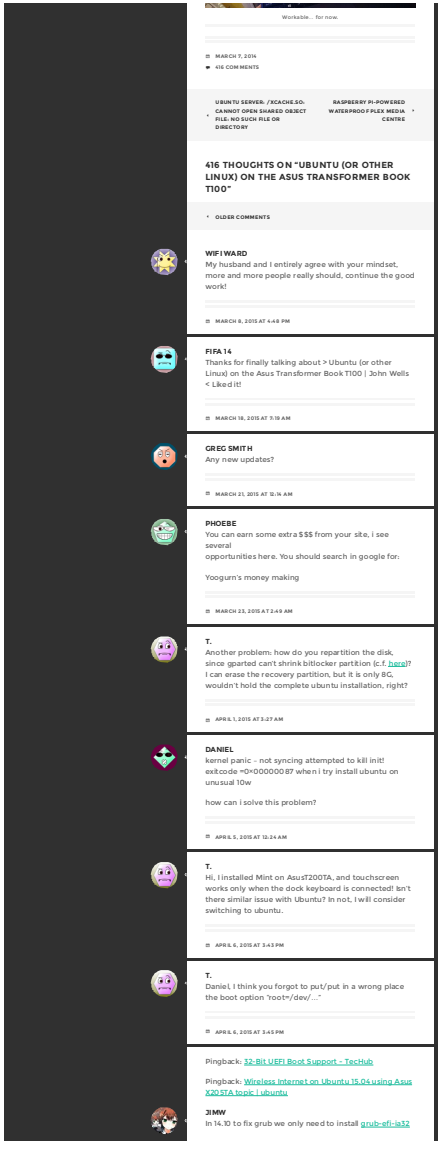
9. Other scripts

I have made some additional small scripts to improve usability.

Dock scripts: These enable/disable the Onboard on-screen keyboard when you undock/dock the tablet. Set up the onscreen keyboard first in System Settings → Accessibility, then the scripts will enable/disable they keyboard for text fields.

Rotate script: This will rotate the screen and touchscreen when run. You can place the desktop entry on your desktop. We can connect the orientation sensor to this later, once that is working.





Workable… for now.

MARCH 7, 2014
416 COMMENTS

« UBUNTU SERVER: /XCACHE.SO: CANNOT OPEN SHARED OBJECT FILE: NO SUCH FILE OR DIRECTORY

RASPBERRY PI-POWERED WATERPROOF FLEX MEDIA CENTRE »

416 THOUGHTS ON "UBUNTU (OR OTHER LINUX) ON THE ASUS TRANSFORMER BOOK T100"

« OLDER COMMENTS

WIFI WARD
My husband and I entirely agree with your mindset, more and more people really should, continue the good work!

MARCH 8, 2015 AT 4:48 PM

FIFA 14
Thanks for finally talking about > Ubuntu (or other Linux) on the Asus Transformer Book T100 | John Wells < Liked it!

MARCH 18, 2015 AT 7:19 AM

GREG SMITH
Any new updates?

MARCH 21, 2015 AT 12:14 AM

PHOEBE
You can earn some extra $$$ from your site, I see several
opportunities here. You should search in google for:

Yoogurn's money making

MARCH 23, 2015 AT 2:49 AM

T.
Another problem: how do you repartition the disk, since gparted can't shrink bitlocker partition (c.f. here)? I can erase the recovery partition, but it is only 8G, wouldn't hold the complete ubuntu installation, right?

APRIL 1, 2015 AT 3:27 AM

DANIEL
kernel panic - not syncing attempted to kill init! exitcode =0×00000087 when i try install ubuntu on unusual 10w

how can i solve this problem?

APRIL 5, 2015 AT 12:24 AM

T.
Hi, I installed Mint on AsusT200TA, and touchscreen works only when the dock keyboard is connected! Isn't there similar issue with Ubuntu? In not, I will consider switching to ubuntu.

APRIL 6, 2015 AT 3:43 PM

T.
Daniel, I think you forgot to put/put in a wrong place the boot option "root=/dev/…"

APRIL 6, 2015 AT 3:45 PM

Pingback: 32-Bit UEFI Boot Support - TecHub

Pingback: Wireless Internet on Ubuntu 15.04 using Asus X20 STA topic | ubuntu

JIMW
In 14.10 to fix grub we only need to install grub-efi-ia32

APR 8, 18, 2015 AT 3:44 PM

Pingback: Linux on ASUS T100ta | SOPEE.NET IT KNOWLEDGE BASE

Pingback: Windowsタブレット EveryPad Pro に入れた microSD に Ubuntu をインストールしてデュアルブートする話み → 失敗 | mogi2fruits どっとねっと

FD
When I try and build into grub-core, I get ./autoshen.sh: line 86: autoreconf: command not found

Tried installing to efi anyway and .../grub-install command isn't found either

JUNE 7, 2015 AT 10:39 AM

EDWARD
Ok, tried this once and had to restore windows when on subsequent reboots I got an error message and the machine kept trying to fix the boot. I won't give up, but one part of the instructions is just a little too vague for me, surprising since the rest is so wonderfully detailed, is the partitioning. I want to eliminate Windows completely and only have Linux. How do you partition for that?

JUNE 23, 2015 AT 3:43 AM

ASK
I do not even know the way I stopped up right here, but I thought this submit was good. I don't know who you
might be but certainly you're going to a famous blogger if you are not already.
Cheers!

JULY 16, 2015 AT 9:08 PM

← OLDER COMMENTS

LEAVE A REPLY

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

You may use these HTML tags and attributes: <a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite=""> <cite> <code> <del datetime=""> <i> <q cite=""> <strike>

POST COMMENT