



**DMP ELECTRONICS INC.**  
**瞻營全電子股份有限公司**

No.15, Wugong 5th Rd. Xinzhuang Dist., New Taipei City, #24890, Taiwan, R.O.C. TEL : 886-2-2298-0770 FAX : 886-2-2299-1883 <http://www.dmp.tw>

# Buildroot for Vortex86EX

(2016/04/15)

# Table of Contents

簡介.....	1
準備.....	1
安裝虛擬機器 ( 可選 ) .....	1
安裝 Buildroot.....	3
配置 Buildroot for VEX-SOM ( 可選 ) .....	4
產生 Linux kernel 與 rootfs image.....	4
使用 Buildroot toolchain for VEX-SOM.....	5
建立 USB 啟動碟 (USBLINUX).....	5
啟動 USBLINUX.....	6
USBLINUX 下存取開發平台的檔案.....	7
安裝 Linux kernel 、 rootfs image 和 bootloader.....	7
完成安裝程序.....	8

## 簡介

Buildroot 是一個製作嵌入式 Linux 系統的工具。它可用來產生嵌入式 Linux 系統的工具鏈、Linux Kernel 及 rootfs 系統。它提供了一個簡單的配置界面，讓使用者可方便客制化嵌入式 Linux 系統內所需要的工具以相關的功能配置。(關於 Buildroot 的詳細說明，請參考 Buildroot 網站上的 The Buildroot user manual。)

在這個文件裡，我們將示範利用 Buildroot 建立一個簡單的 Embedded Linux，並且利用 USB 啟動碟安裝這個 Embedded Linux 至 Vortex86EX 平台。

## 準備

一台 PC 工作機，並且連接上網路。

一個 Vortex86EX VEX-SOM 模組和底板 (這邊選用的是 ICOP VEX-6254)。

```
ICOP VEX-6254 的簡要規格如下：  
DMP SoC Vortex86EX  
128MB DDR3 onboard  
Support microSD card or 512MB eMMC
```

一個 USB 隨身碟。

一個 microSD 卡。

(部分型號的 ICOP VEX-6254 只有支援 eMMC，請直接使用板子上的 eMMC 替代 microSD 卡。)

一條網路線，將 VEX-SOM 連接上與 PC 工作機互通的網路。

## 安裝虛擬機器 (可選)

Buildroot 需在 Linux 平台下工作。若是工作環境是在 Windows 底下，請先安裝虛擬機器，然後在虛擬機器上安裝 Linux distribution。我們建議安裝 Debian 7(or later)作為 Linux 開發平台。

1. 下載並安裝虛擬機器 (此處以 VirtualBox 為例)。

```
VirtualBox Website: <https://www.virtualbox.org/>
```

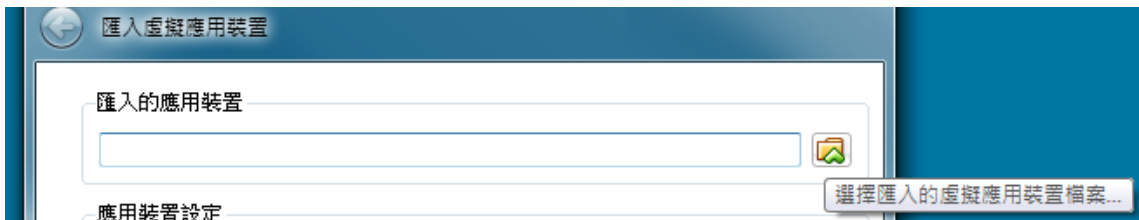
2. 下載 VM 應用裝置檔案。

```
ftp://vxex:gc301@ftp.dmp.com.tw/Software/Linux/debian8.3.0_vexsom-  
standard_develop_Env.ova
```

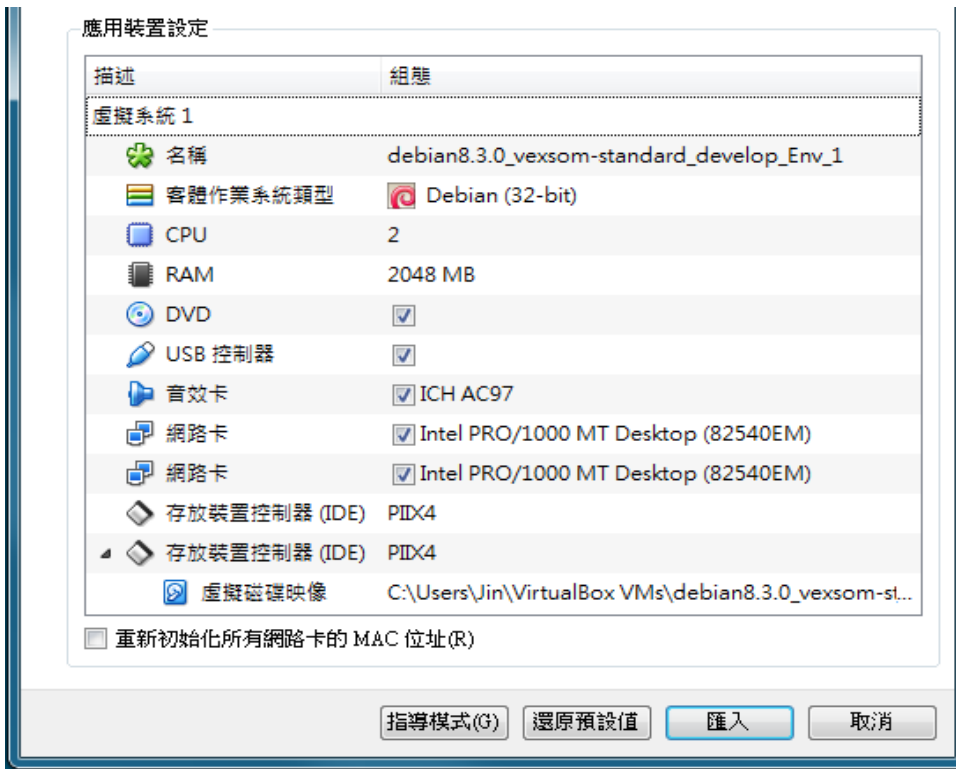
3. 開啟 [ Oracle VM VirtualBox ] 應用程式，選擇 [ 檔案 ] → [ 匯入應用裝置(I)... ] 。



4. 在 [ 匯入的應用裝置 ] 欄位右側，點擊 [ 選擇匯入的虛擬應用裝置檔案... ] 圖示，點選下載下來的 debian8.3.0\_vexsom-standard\_develop\_Env.ova 檔案，按 [ 開啟舊檔 ] 。



5. 依照目前執行 VirtualBox 的電腦規格及個人需求，修改虛擬機器的各項設定 ( 如名稱、CPUs、RAM 等 )，或使用原預設值直接點擊 [ 匯入 ] 。



- 匯入完成後，會在 [ Oracle VM VirtualBox 管理員 ] 右側 VM 列表出現剛才匯入的虛擬機器。點擊該機器項目即可開啟。登入 user 後可直接進行 make ( 請參考 " 產生 Linux kernel 與 rootfs image " 章節 ) 。



## 安裝 Buildroot

- 範例中的工作目錄都設定在 \$HOME 。

```
$ cd $HOME
```

- 下載 buildroot-2015.02.tar.gz 檔案。

```
$ wget ftp://vxex:gc301@ftp.dmp.com.tw/Software/Linux/buildroot-2015.02.tar.gz
```

- 下載 buildroot-vortex86ex.tar.gz 檔案。

```
$ wget ftp://vxex:gc301@ftp.dmp.com.tw/Software/Linux/buildroot-vortex86ex.tar.gz
```

- 解壓縮下載的檔案。

```
$ tar -xf buildroot-2015.02.tar.gz  
$ tar -xf buildroot-vortex86ex.tar.gz
```

- 更改目錄名稱。

```
$ mv buildroot-2015.02 buildroot-2015.02-vexsom
```

- 定義必要的環境變數。

```
$ export BR2_EXTERNAL=$HOME/buildroot-external  
$ export BR2_DL_DIR=$HOME/downloads
```

7. 進入 buildroot-2015.02-vexsom 目錄後，導入 vexsom\_defconfig。

```
$ cd buildroot-2015.02-vexsom
$ make vexsom_defconfig
```

## 配置 Buildroot for VEX-SOM ( 可選 )

Buildroot 提供了許多的配置選項，使用者可透過 make menuconfig 進入配置選單，在配置選單可進一步進入 Target packages 選擇所需的 package。若不需要配置 Buildroot，則請直接參考下一章節。

```
$ make menuconfig
```

Linux kernel 的配置，請執行 make linux-menuconfig 進行相關的配置。

```
$ make linux-menuconfig
```

Busybox 的配置，請執行 make busybox-menuconfig 進行相關的配置。

```
$ make busybox-menuconfig
```

若需要添加額外的檔案至 rootfs 裡，可以直接將檔案複製到 fs\_overlay 的目錄下，這樣當 make 完成後，Buildroot 會自動將 fs\_overlay 的檔案複製或覆蓋原始的 rootfs 檔案。

這裡示範將程式 hello 安裝到 rootfs 的 /usr/bin/ 目錄。

```
$ mkdir -p $HOME/buildroot-external/board/icop/vexsom/fs_overlay/usr/bin
$ cp hello $HOME/buildroot-external/board/icop/vexsom/fs_overlay/usr/bin/
```

## 產生 Linux kernel 與 rootfs image

配置完成後，就可以開始進行 make。make 過程需要等待一些時間，待 Buildroot 系統完成所有步驟後，Image 檔案會放在 output/images/ 目錄底下。

```
$ make
$ ls output/image
bzImage rootfs.tar rootfs.tar.gz
```

在 output/images 目錄下，使用者會需要兩個檔案，這兩個檔案分別是 Linux kernel 和 rootfs image：

'bzImage': Linux kernel image

'rootfs.tar.gz' : rootfs image, contains all files for root filesystem

## 使用 Buildroot toolchain for VEX-SOM

Buildroot 完成 make 後，會將相關的 toolchain 檔案放在 output/host/ 目錄底下，例如 gcc 的路徑是 output/host/usr/bin/i686-buildroot-linux-gnu-gcc，而 toolchain sysroot 會在 output/host/usr/i686-buildroot-linux-gnu/sysroot/（Buildroot 有特別做一個 link 在 output/staging/）。

這裡示範使用 Buildroot toolchain 來編譯 hello.c 程式，並且：

1. 設定 PATH

```
$ export PATH=$PATH:$HOME/buildroot-2015.02-vexsom/output/host/usr/bin
```

2. 建立和編譯 hello.c

```
$ cat >hello.c <<EOF
#include <stdio.h>
int main()
{
    printf("Hello Buildroot!\n");
    return 0;
}
EOF

$ i686-buildroot-linux-gnu-gcc -o hello hello.c
```

3. 將 hello 程式放到 fs\_overlay 下，並且重新 make。

```
$ mkdir -p $HOME/buildroot-external/board/icop/vexsom/fs_overlay/usr/bin
$ cp hello $HOME/buildroot-external/board/icop/vexsom/fs_overlay/usr/bin/
$ make
```

4. make 完成後，hello 程式就會在 target/usr/bin/ 目錄下以及 rootfs.tar.gz 檔案裡。

## 建立 USB 啟動碟 (USBLINUX)

目前已經完成了 VEX-SOM 的 Linux kernel 和 rootfs image，接下來要討論如何將 image 檔案安裝

到 VEX-SOM 上。

這裡使用的方式是建立一個 USB 啟動碟，可在 VEX-SOM 上啟動，並且用來安裝上個步驟建立好的 Linux kernel 以及 rootfs image。

1. 進入工作目錄 \$HOME。

```
$ cd $HOME
```

2. 解壓縮 buildroot-2015.02.tar.gz。

```
$ tar -xf buildroot-2015.02.tar.gz
```

3. 更改目錄名稱。

```
$ mv buildroot-2015.02 buildroot-2015.02-usblinux
```

4. 定義必要的環境變數。

```
$ export BR2_EXTERNAL=$HOME/buildroot-external  
$ export BR2_DL_DIR=$HOME/downloads
```

5. 進入 buildroot-2015.02-usblinux 後，導入 usblinux 的 defconfig，並且進行 make。

```
$ cd buildroot-2015.02-usblinux  
$ make usblinux_defconfig  
$ make
```

6. 待 make 完成後，在 output/images 目錄下，會看到一個 usblinux.iso 的檔案。

```
$ ls output/images  
bzImage  rootfs.cpio  rootfs.cpio.gz  syslinux  usblinux.iso
```

7. 這時插入 USB 隨身碟，掛上虛擬機器後，就可以透過 dd 程式直接將 usblinux.iso 寫入 USB 隨身碟。(假設 USB 隨身碟是 /dev/disk/by-label/usbdrive)

```
$ dd if=output/images/usblinux.iso of=/dev/disk/by-label/usbdrive bs=4k
```

8. 完成後，請取下 USB 隨身碟。

## 啟動 USBLINUX

請在 ICOP VEX-6254 上插入 microSD 卡，以及 USB 隨身碟後，啟動系統電源。當 USB 隨身碟系統啟動完成後，會自動設定網路 IP, Netmask 如下，並且啟動 SSH server。

```
IP      : 192.168.100.100
```



```
Netmask: 255.255.255.0
```

若開發平台與 VEX-6254 在同一個網路上，這時就可以在開發平台上透過 SSH client 登入 USBLINUX。

```
$ ssh root@192.168.100.100  
( 密碼 : vortex86ex )  
root@usblinux:~#
```

## USBLINUX 下存取開發平台的檔案

USBLINUX 沒有支援 NFS，取而代之支援了 SSHFS 和 CurlFtpFS，透過這兩個程式使用者可以分別將 SSH server 或 FTP server 掛載到某個目錄。

```
(hostip 為開發平台的 IP)  
root@usblinux:~# mkdir /mnt/host  
root@usblinux:~# sshfs user@hostip:/path /mnt/host  
or  
root@usblinux:~# curlftpfs ftp://hostip/path /mnt/host
```

## 安裝 Linux kernel、rootfs image 和 bootloader

這裡會使用 USBLINUX 上的自動安裝程式做安裝。安裝程式會自動將 microSD card 規劃成單一 partition，並且在這個 partition 上建立 ext3 filesystem，安裝 Linux kernel、root filesystem，最後安裝 extlinux (syslinux) bootloader。

在進行自動安裝之前，需要先將 Linux kernel 和 rootfs image 複製到 ~/ 目錄底下。

以下假設開發平台上的 output/images 目錄，掛載到 USBLINUX 下的 /mnt/host 目錄。

```
root@usblinux:~# cp /mnt/host/bzImage ~/  
root@usblinux:~# cp /mnt/host/rootfs.tar.gz ~/
```

另外安裝程式還需要指定目標磁碟的磁碟代號。一般而言 microSD 的磁碟代號會是 '/dev/sda' (Kernel 第一個抓到的 Block device)。但是不同的平台有可能會有不同的磁碟代號，建議可用 'fdisk -l' 或 'lsblk -S' 指令來確認。

最後執行 make-target 自動完成所有的安裝程序。

```
root@usblinux:~# /opt/bin/make-target /dev/sda
```

## 完成安裝程序

待 make-target 執行完畢後，microSD 上就是可開機的 Embedded Linux 的系統。這時就可以將 USB 啟動碟移除，並且將電源重新啟動，進入 Buildroot 所打造的 Linux 環境。

若您有任何問題，歡迎 email 聯絡 <soc@dmp.com.tw>。