Embedded Linux

Splash screen for Embedded Linux 101: How to customize your boot sequence

Pierre-Jean Texier

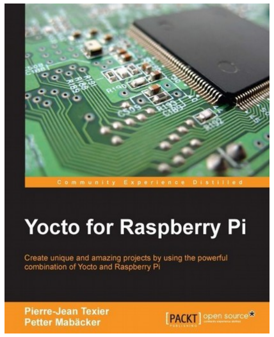
Live Embedded 2021

MADIC group

Pierre-Jean Texier

- Embedded Linux Engineer **LAFON** (part of Madic group)

- 30 yo (~ 31) – Father of 2 -
- **FOSS** enthusiast
- <u>Contributions</u> : U-Boot, Kernel Linux, Yocto/OE, Buildroot …
- Co-author of "*Yocto for Raspberry Pi*" and author for *GNU/Linux magazine France* and Open silicium (RIP)


Yocto for Raspberry Pi
Create unique and amazing projects by using the powerful combination of Yocto and Raspberry Pi
Pierre-Jean Texier
Petter Mabäcker
PACKT open source

Agenda

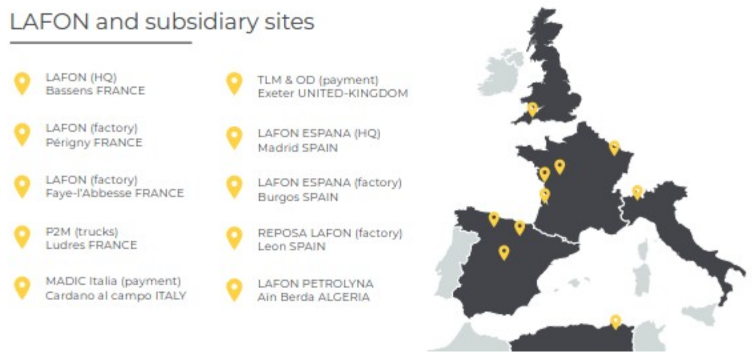
- About Us : LAFON
- Splash Screen ?
- How to add a splashscreen in U-Boot ?
- How to change the boot logo of the Linux kernel ?
- Dynamic splash screen in userspace area ?
- Summary

MADIC group

LAFON

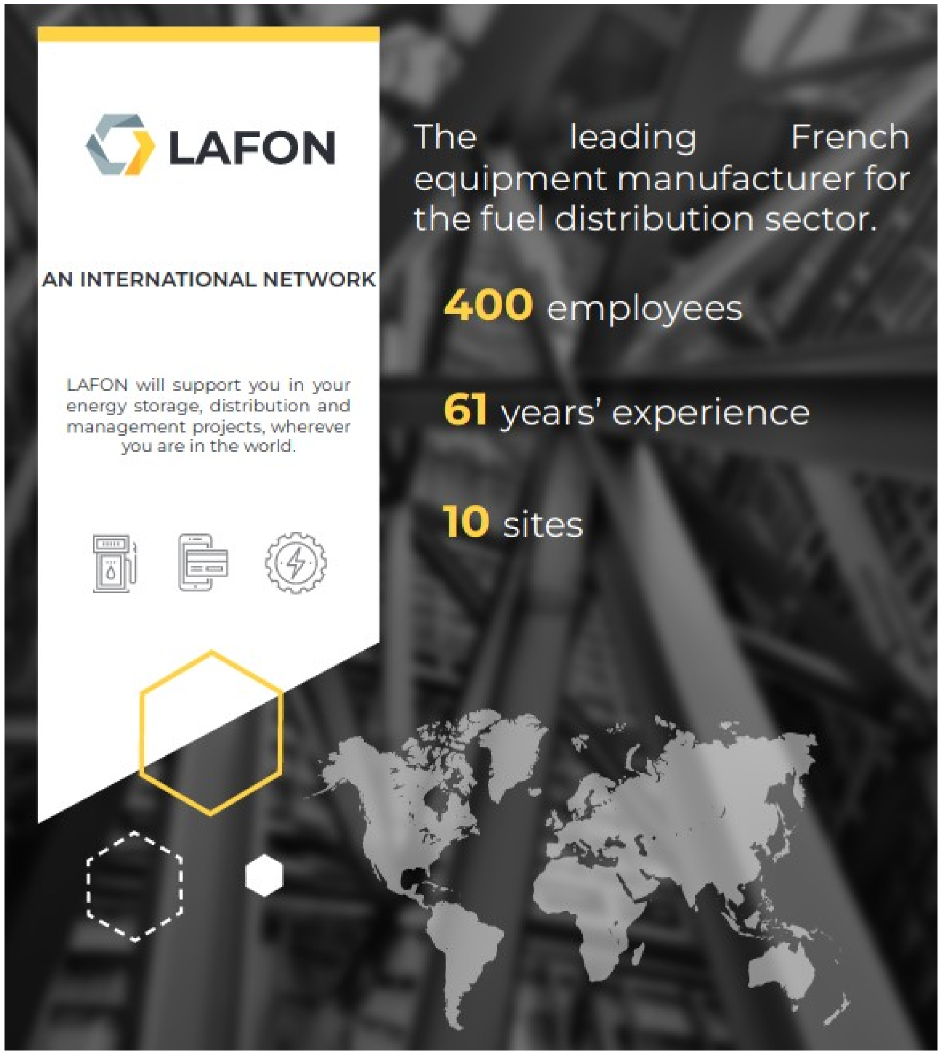
About us : LAFON

- **Founded in Bordeaux in 1959**
- **part of the MADIC group since 2006**
- **a « *leading industrial group which specialises in Energy* »**
- **Sites :**



LAFON and subsidiary sites

LAFON (HQ) Bassens FRANCE	TLM & OD (payment) Exeter UNITED-KINGDOM
LAFON (factory) Périgny FRANCE	LAFON ESPANA (HQ) Madrid SPAIN
LAFON (factory) Faye-l'Abbesse FRANCE	LAFON ESPANA (factory) Burgos SPAIN
P2M (trucks) Ludres FRANCE	REPOSA LAFON (factory) Leon SPAIN
MADIC Italia (payment) Cardano al campo ITALY	LAFON PETROLYNA Aïn Berda ALGERIA



LAFON

AN INTERNATIONAL NETWORK

LAFON will support you in your energy storage, distribution and management projects, wherever you are in the world.

The leading French equipment manufacturer for the fuel distribution sector.

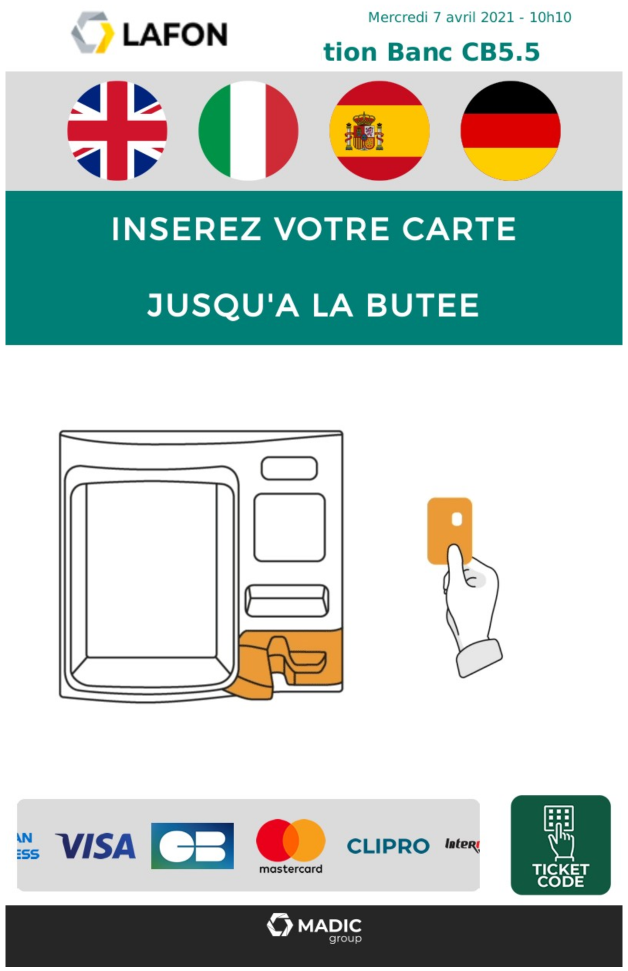
400 employees

61 years' experience

10 sites

LAFON : ‘payment’ service

- **Service:**
  - Oriented payments
  - Oriented system (*embedded Linux*)
- **Requirements … :**
  - PCI DSS
  - PCI P2PE
  - GIE Carte Bancaire
  - …
- **Security … :**
  - Trustzone
  - Secure Boot, …
  - …
- **Open Source**
  - Qt (*GUI*)
  - Yocto/OpenEmbedded (*build system*)
  - cURL, libevent, libxml2, …



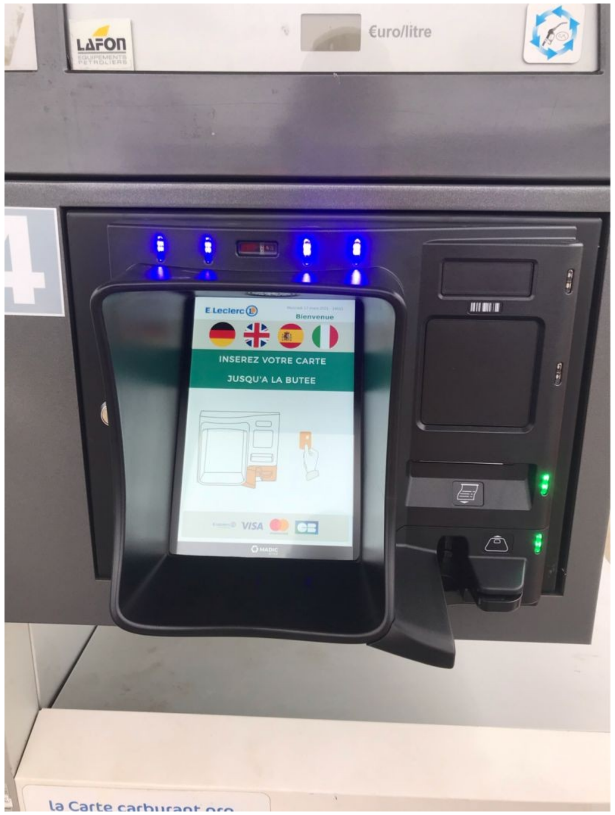
Mercredi 7 avril 2021 - 10h10

tion Banc CB5.5

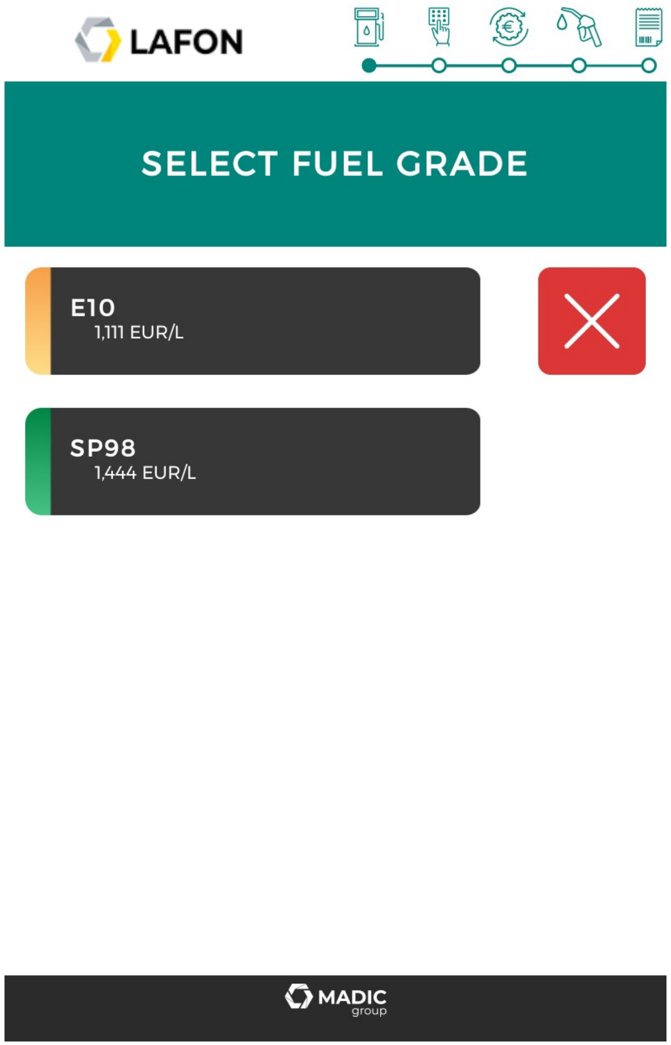
INSEREZ VOTRE CARTE

JUSQU'A LA BUTEE

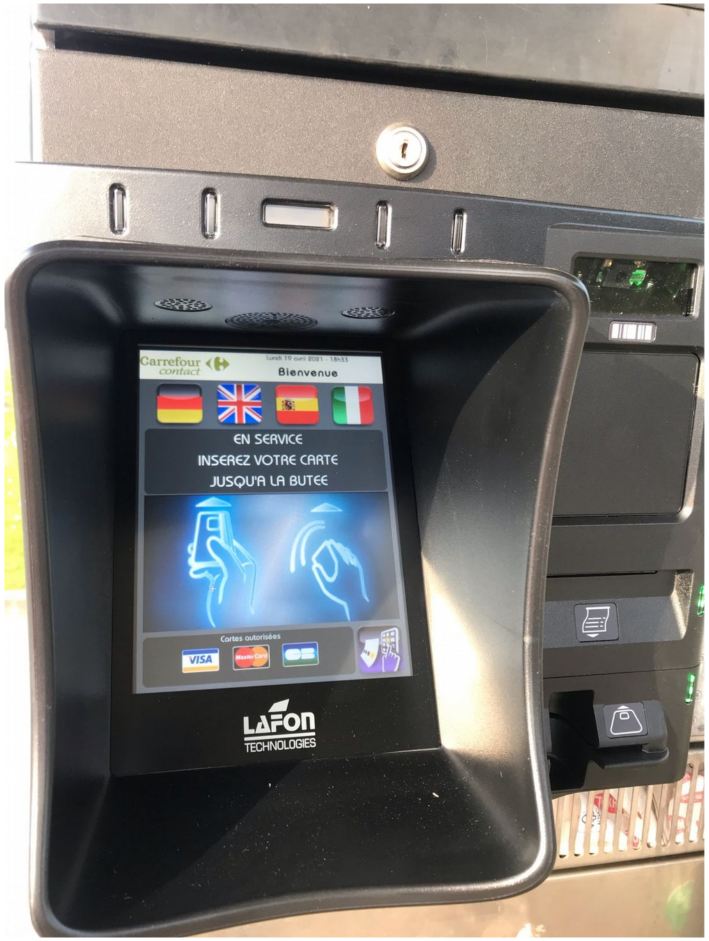
LAFON : APL3.5



Specifications



LAFON : APL3 « legacy »



Introduction

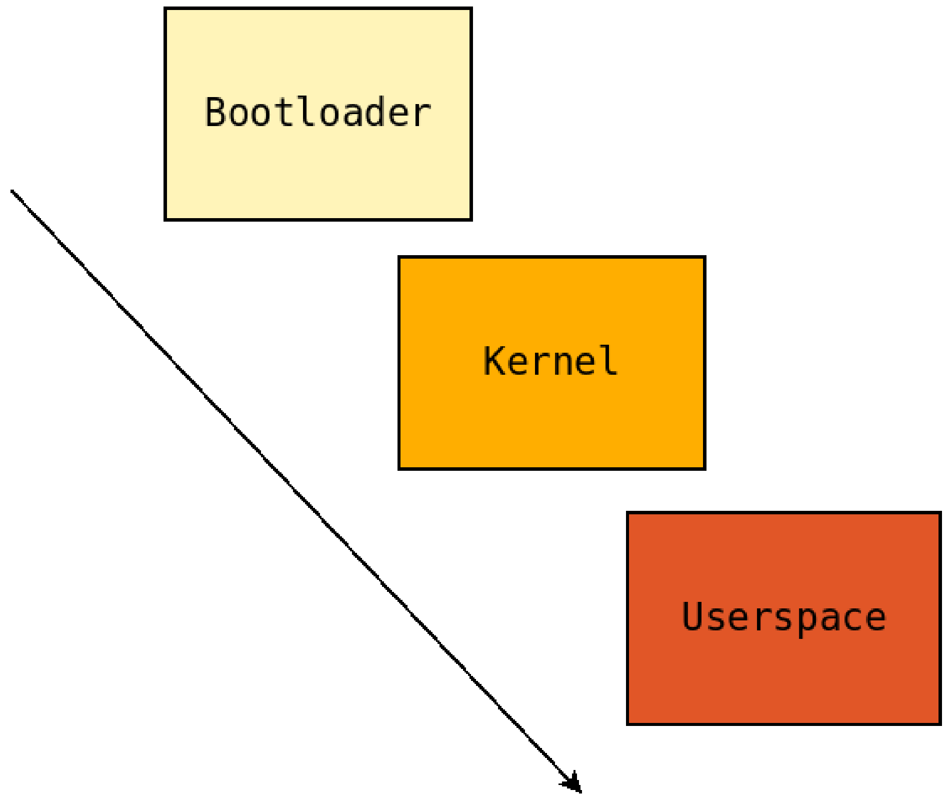
Why a splash screen ?

- Customer
- Responsiveness
- Vendor (customization)

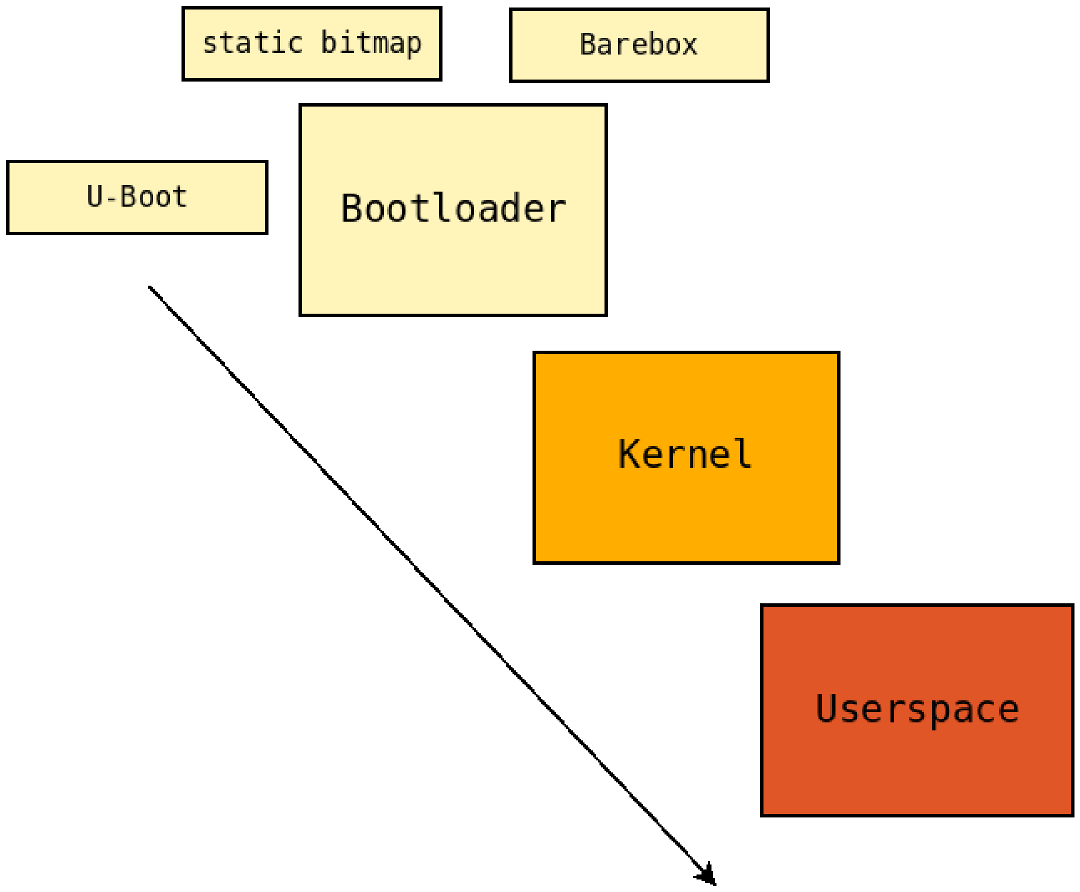
LAFON

MADIC group

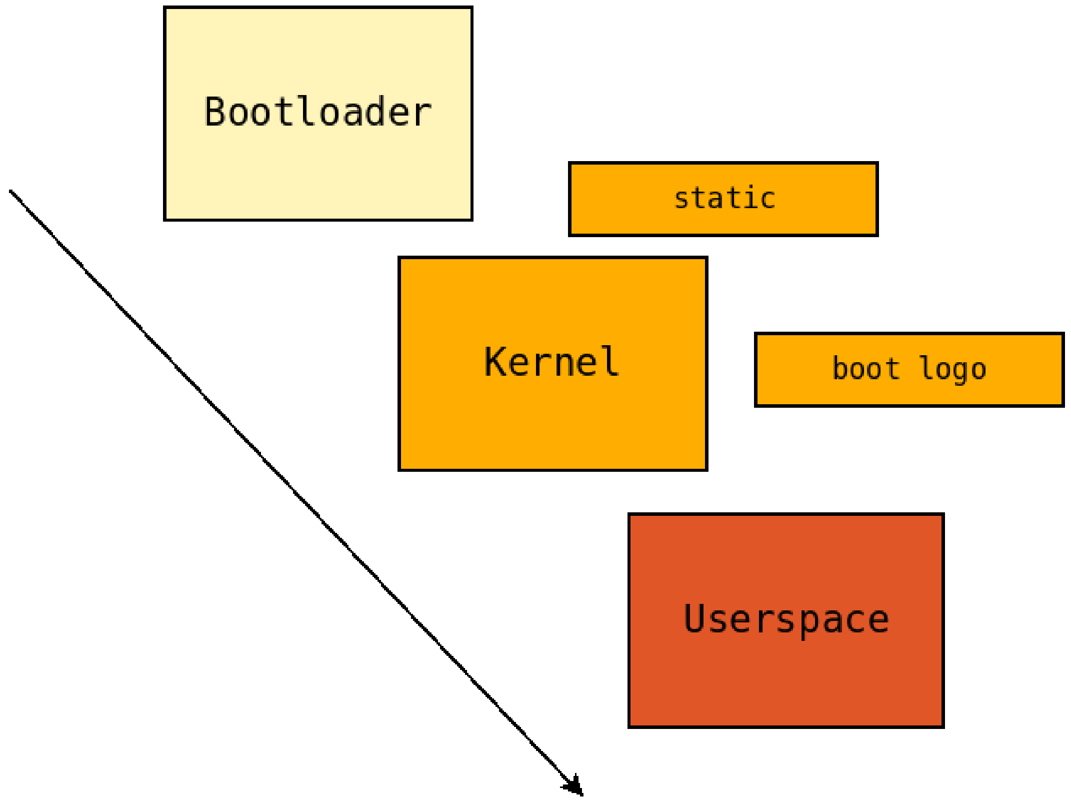
Where ?



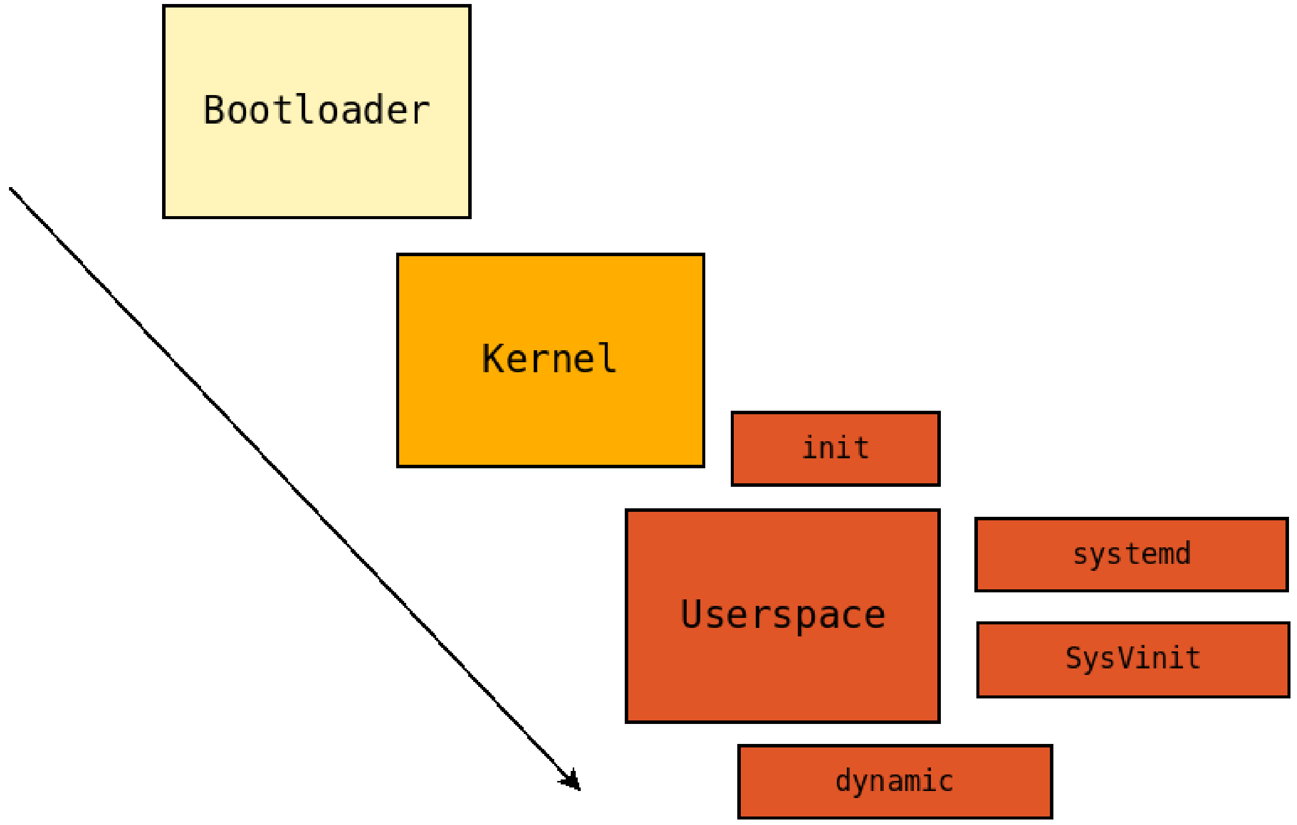
Where ?



Where ?



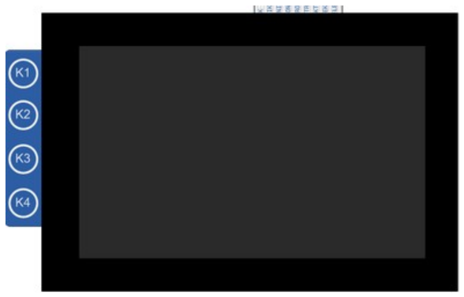
Where ?


Bootloader
Kernel
init
Userspace
systemd
SysVinit
dynamic

U-Boot

U-Boot : Setup

- ***Microchip sama5d27-som1-ek1*** + PDA touchscreen display



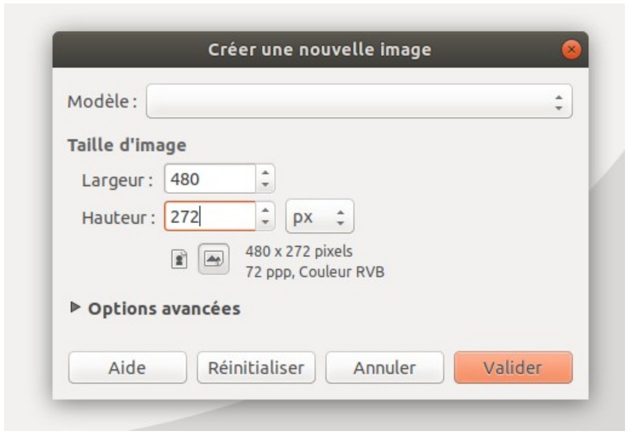
- Image generated with Yocto/OE & Kas:
  - Release : dunfell
  - U-Boot version « ***u-boot-2021.04-at91*** »

U-Boot : preparation

- U-Boot can enable support for static bitmap images …
- So, we need to create (first) our « **splashscreen** » file

U-Boot : preparation

- U-Boot can enable support for static bitmap images …
- So, we need to create (first) our « **splashscreen** » file
- for the *capacitive 4.3 inch display* with a resolution of 480x272 :



Créer une nouvelle image

Modèle :

Taille d'image

Largeur : 480

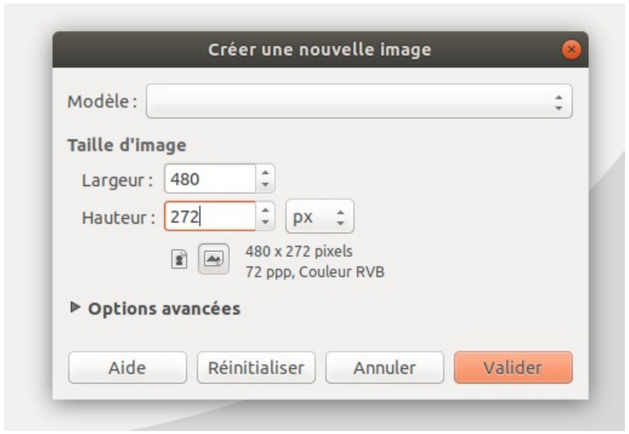
Hauteur : 272 px

480 x 272 pixels
72 ppp, Couleur RVB

▶ Options avancées

U-Boot : preparation

- U-Boot can enable support for static bitmap images …
- So, we need to create (first) our « **splashscreen** » file
- for the *capacitive 4.3 inch display* with a resolution of 480x272 :

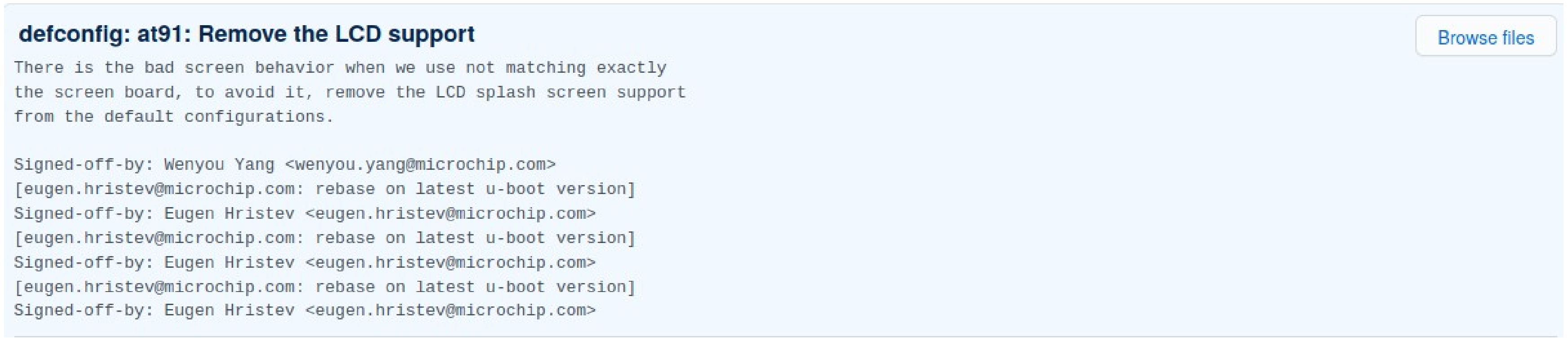


- And convert our pngfile to bmp (*8-bit depth*)

```
$: pngtopnm madic.png | ppmquant 256 | ppmtobmp -bpp 8 > madic.bmp
ppmquant: making histogram...
ppmquant: 825 colors found
ppmquant: choosing 256 colors...
ppmquant: mapping image to new colors...
ppmtobmp: analyzing colors...
ppmtobmp: 176 colors found
ppmtobmp: Writing 8 bits per pixel with a color pallette
```

U-Boot : « *BMP* »

- By adding **`CONFIG_CMD_BMP`** option, it is possible to manipulate and display a bmp file
- … but we need also **`CONFIG_DM_VIDEO`** and **`CONFIG_ATMEL_HLCD`** options
- **LCD support** is enable upstream but disabled on Atmel's fork



defconfig: at91: Remove the LCD support

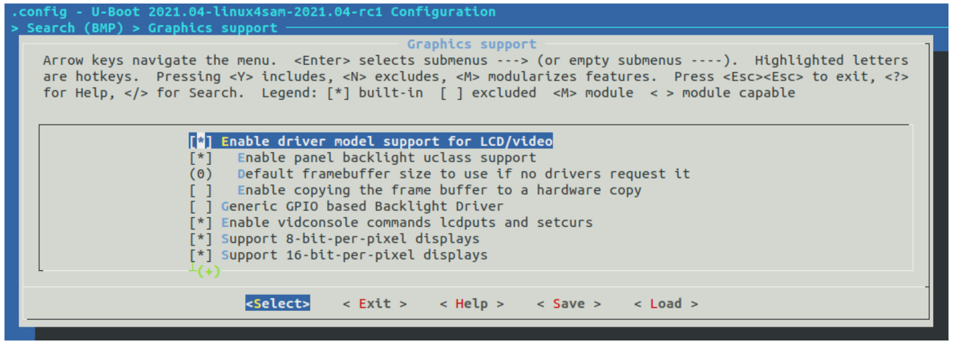
```
There is the bad screen behavior when we use not matching exactly
the screen board, to avoid it, remove the LCD splash screen support
from the default configurations.

Signed-off-by: Wenyou Yang <wenyou.yang@microchip.com>
[eugen.hristev@microchip.com: rebase on latest u-boot version]
Signed-off-by: Eugen Hristev <eugen.hristev@microchip.com>
[eugen.hristev@microchip.com: rebase on latest u-boot version]
Signed-off-by: Eugen Hristev <eugen.hristev@microchip.com>
[eugen.hristev@microchip.com: rebase on latest u-boot version]
Signed-off-by: Eugen Hristev <eugen.hristev@microchip.com>
```

`https://github.com/linux4sam/u-boot-at91/commit/536162a3489307c32f398fbc501ebbecdc584830`

- So, we need to revive this functionnality !

U-Boot : « *BMP* »

- « **`make menuconfig`** » :

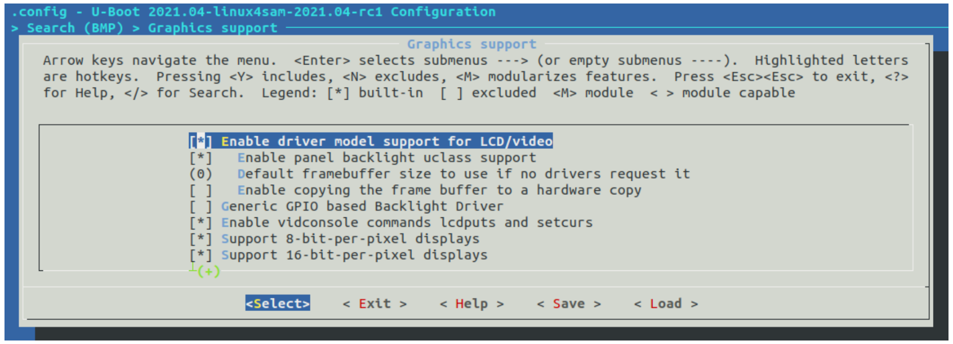


```
.config - U-Boot 2021.04-linux4sam-2021.04-rc1 Configuration
> Search (BMP) > Graphics support
                                Graphics support
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters
  are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?>
  for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                      [*] Enable driver model support for LCD/video
                      [*]   Enable panel backlight uclass support
                      (0)   Default framebuffer size to use if no drivers request it
                      [ ]   Enable copying the frame buffer to a hardware copy
                      [ ] Generic GPIO based Backlight Driver
                      [*] Enable vidconsole commands lcdputs and setcurs
                      [*] Support 8-bit-per-pixel displays
                      [*] Support 16-bit-per-pixel displays
                      ↓(+)

                   <Select>    < Exit >    < Help >    < Save >    < Load >
```

Device Drivers → Graphics support → Enable DM Video

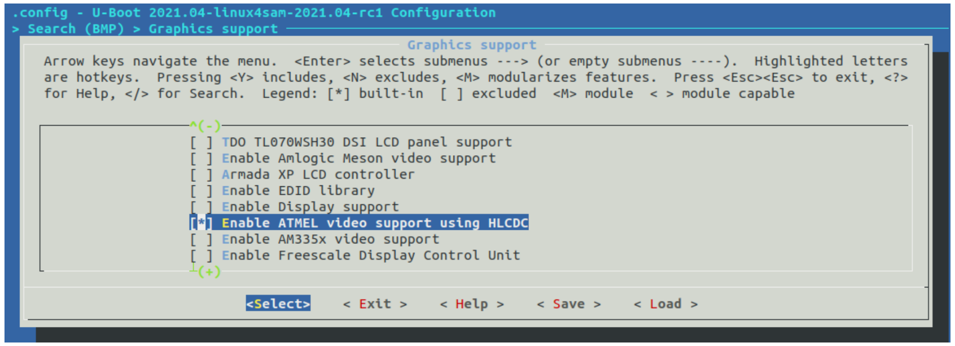
U-Boot : « *BMP* »

- « **`make menuconfig`** » :



```
.config - U-Boot 2021.04-linux4sam-2021.04-rc1 Configuration
> Search (BMP) > Graphics support
                                   Graphics support
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters
  are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?>
  for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                   [*] Enable driver model support for LCD/video
                   [*]   Enable panel backlight uclass support
                   (0)   Default framebuffer size to use if no drivers request it
                   [ ]   Enable copying the frame buffer to a hardware copy
                   [ ] Generic GPIO based Backlight Driver
                   [*] Enable vidconsole commands lcdputs and setcurs
                   [*] Support 8-bit-per-pixel displays
                   [*] Support 16-bit-per-pixel displays
                   ↓(+)

                     <Select>    < Exit >    < Help >    < Save >    < Load >
```

Device Drivers → Graphics support → Enable DM Video



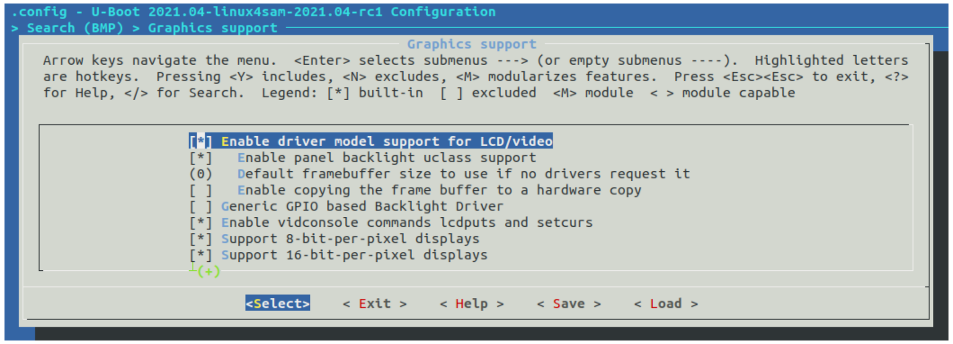
```
.config - U-Boot 2021.04-linux4sam-2021.04-rc1 Configuration
> Search (BMP) > Graphics support
                                   Graphics support
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters
  are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?>
  for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                   ^(-)
                   [ ] TDO TL070WSH30 DSI LCD panel support
                   [ ] Enable Amlogic Meson video support
                   [ ] Armada XP LCD controller
                   [ ] Enable EDID library
                   [ ] Enable Display support
                   [*] Enable ATMEL video support using HLCDC
                   [ ] Enable AM335x video support
                   [ ] Enable Freescale Display Control Unit
                   ↓(+)

                     <Select>    < Exit >    < Help >    < Save >    < Load >
```

Device Drivers → Graphics support → Enable Atmel Video

MADIC group

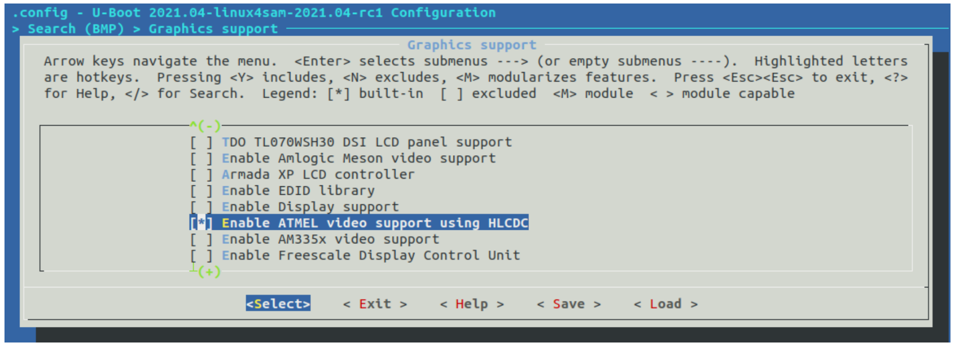
U-Boot : « *BMP* »

- « **`make menuconfig`** » :



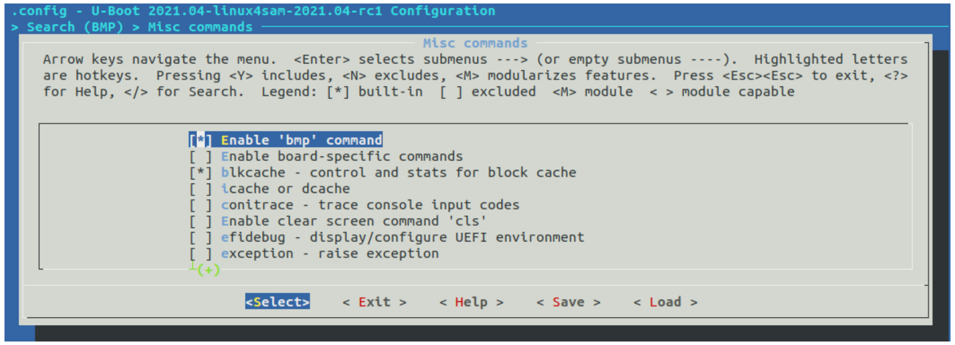
```
.config - U-Boot 2021.04-linux4sam-2021.04-rc1 Configuration
> Search (BMP) > Graphics support
                              Graphics support
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters
  are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?>
  for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                   [*] Enable driver model support for LCD/video
                   [*]   Enable panel backlight uclass support
                   (0)   Default framebuffer size to use if no drivers request it
                   [ ]   Enable copying the frame buffer to a hardware copy
                   [ ] Generic GPIO based Backlight Driver
                   [*] Enable vidconsole commands lcdputs and setcurs
                   [*] Support 8-bit-per-pixel displays
                   [*] Support 16-bit-per-pixel displays
                   ↓(+)

                   <Select>    < Exit >    < Help >    < Save >    < Load >
```

Device Drivers → Graphics support → Enable DM Video



```
.config - U-Boot 2021.04-linux4sam-2021.04-rc1 Configuration
> Search (BMP) > Graphics support
                              Graphics support
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters
  are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?>
  for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                   ↑(-)
                   [ ] TDO TL070WSH30 DSI LCD panel support
                   [ ] Enable Amlogic Meson video support
                   [ ] Armada XP LCD controller
                   [ ] Enable EDID library
                   [ ] Enable Display support
                   [*] Enable ATMEL video support using HLCDC
                   [ ] Enable AM335x video support
                   [ ] Enable Freescale Display Control Unit
                   ↓(+)

                   <Select>    < Exit >    < Help >    < Save >    < Load >
```

Device Drivers → Graphics support → Enable Atmel Video



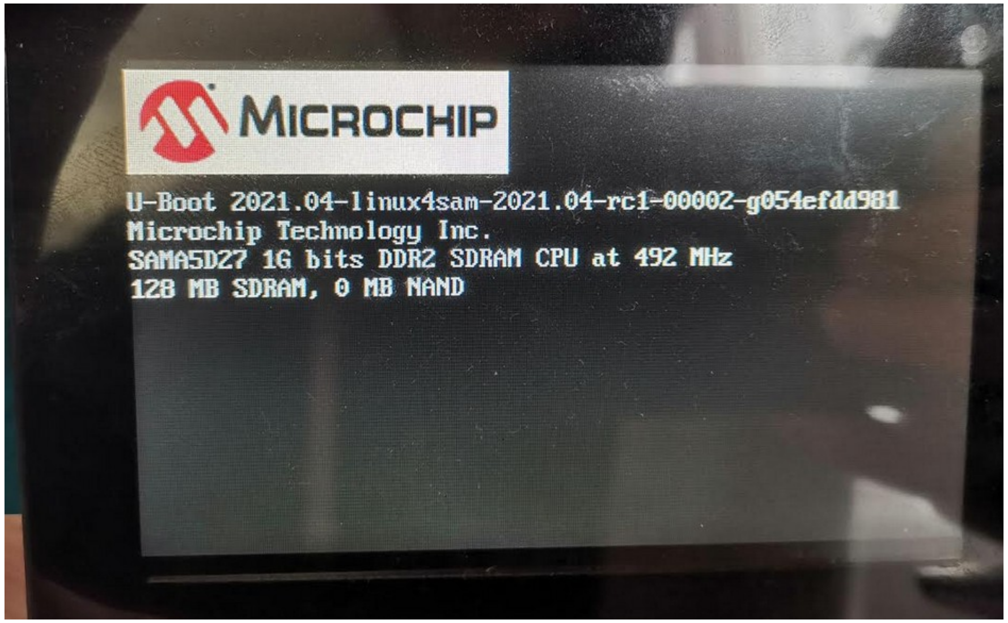
```
.config - U-Boot 2021.04-linux4sam-2021.04-rc1 Configuration
> Search (BMP) > Misc commands
                              Misc commands
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters
  are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?>
  for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                   [*] Enable 'bmp' command
                   [ ] Enable board-specific commands
                   [*] blkcache - control and stats for block cache
                   [ ] icache or dcache
                   [ ] conitrace - trace console input codes
                   [ ] Enable clear screen command 'cls'
                   [ ] efidebug - display/configure UEFI environment
                   [ ] exception - raise exception
                   ↓(+)

                   <Select>    < Exit >    < Help >    < Save >    < Load >
```

Command line interface → Misc commands → Enable bmp command

MADIC group

U-Boot : Vendor logo

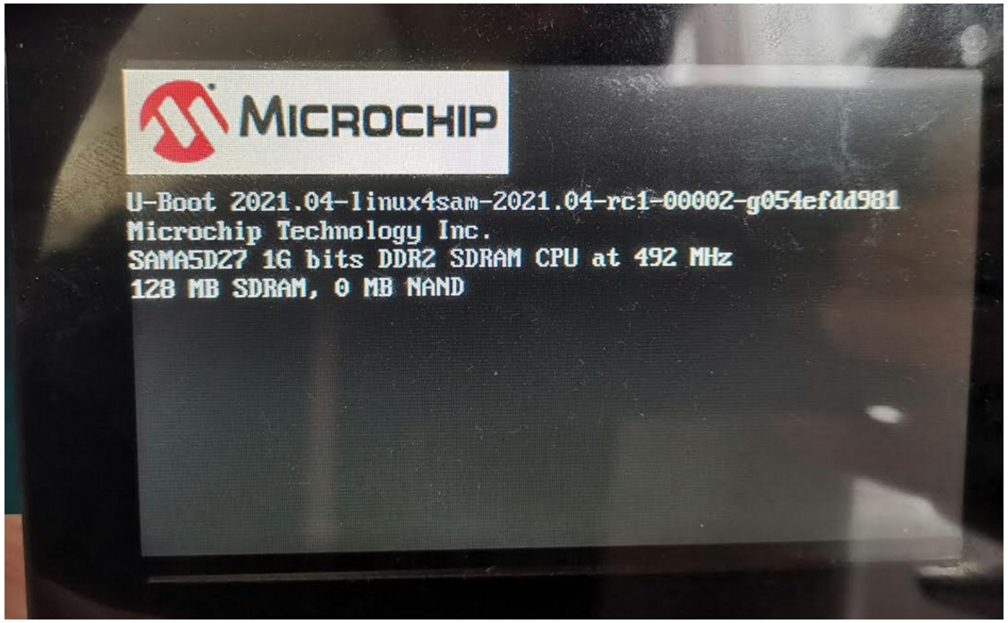
- Let’s try the new **`u-boot.bin`** !



```
U-Boot 2021.04-linux4sam-2021.04-rc1-00002-g054efdd981
Microchip Technology Inc.
SAMA5D27 1G bits DDR2 SDRAM CPU at 492 MHz
128 MB SDRAM, 0 MB NAND
```

U-Boot : Vendor logo

- Let's try the new **`u-boot.bin`** !



```
U-Boot 2021.04-linux4sam-2021.04-rc1-00002-g054efdd981
Microchip Technology Inc.
SAMA5D27 1G bits DDR2 SDRAM CPU at 492 MHz
128 MB SDRAM, 0 MB NAND
```

- The **Vendor logo** (if **`CONFIG_DM_VIDEO`** is set) is :

  - stored in **U-Boot** binary itself
  - Defined (statically) in *`<U-Boot>/lib/at91/microchip_logo_8bpp.h`*
  - And used by the function *`at91_video_show_board_info()`*

U-Boot : « *BMP* »

- The function **`video_bmp_display()`** is used to display the BMP file:

```
microchip_logo_info(&logo_info);
ret = video_bmp_display(dev, logo_info.logo_addr,
        logo_info.logo_x_offset,
         logo_info.logo_y_offset, false);
if (ret)
       return ret;
```

```
void microchip_logo_info(vidinfo_t *info)
{
       info->logo_width = MICROCHIP_LOGO_8BPP_WIDTH;
       info->logo_height = MICROCHIP_LOGO_8BPP_HEIGHT;
       info->logo_x_offset = MICROCHIP_LOGO_8BPP_X_OFFSET;
       info->logo_y_offset = MICROCHIP_LOGO_8BPP_X_OFFSET;
       info->logo_addr = (u_long)microchip_logo_8bpp;
}
```

U-Boot : « *BMP* »

- The function `video_bmp_display()` is used to display the BMP file:

```
microchip_logo_info(&logo_info);
ret = video_bmp_display(dev, logo_info.logo_addr,
        logo_info.logo_x_offset,
        logo_info.logo_y_offset, false);
if (ret)
       return ret;
```

```
void microchip_logo_info(vidinfo_t *info)
{
       info->logo_width = MICROCHIP_LOGO_8BPP_WIDTH;
       info->logo_height = MICROCHIP_LOGO_8BPP_HEIGHT;
       info->logo_x_offset = MICROCHIP_LOGO_8BPP_X_OFFSET;
       info->logo_y_offset = MICROCHIP_LOGO_8BPP_X_OFFSET;
       info->logo_addr = (u_long)microchip_logo_8bpp;
}
```

- So, not very convenient for our use case … If we need to remove such informations at startup, the following changes should be made on board definition file:

```
diff --git a/board/atmel/sama5d27_som1_ek/sama5d27_som1_ek.c b/board/atmel/sama5d27_som1_ek/sama5d27_som1_ek.c
index 1b7d946b50..8516eadba1 100644
--- a/board/atmel/sama5d27_som1_ek/sama5d27_som1_ek.c
+++ b/board/atmel/sama5d27_som1_ek/sama5d27_som1_ek.c
@@ -32,7 +32,7 @@ static void board_usb_hw_init(void)
 int board_late_init(void)
 {
 #ifdef CONFIG_DM_VIDEO
-      at91_video_show_board_info();
+      /* at91_video_show_board_info(); */
 #endif
       at91_pda_detect();
       return 0;
```

U-Boot : « *BMP* »

- After this trivial modification, we are able to play with our bmp file (without the Vendor Logo) :

```
=> fatload mmc 0:1 0x27000000 madic.bmp
131638 bytes read in 26 ms (4.8 MiB/s)
```

MADIC group

U-Boot : « *BMP* »

- After this trivial modification, we are able to play with our bmp file (without the Vendor Logo) :

```
=> fatload mmc 0:1 0x27000000 madic.bmp
131638 bytes read in 26 ms (4.8 MiB/s)
```

```
=> bmp info 0x27000000
Image size    : 480 x 272
Bits per pixel: 8
Compression   : 0
```

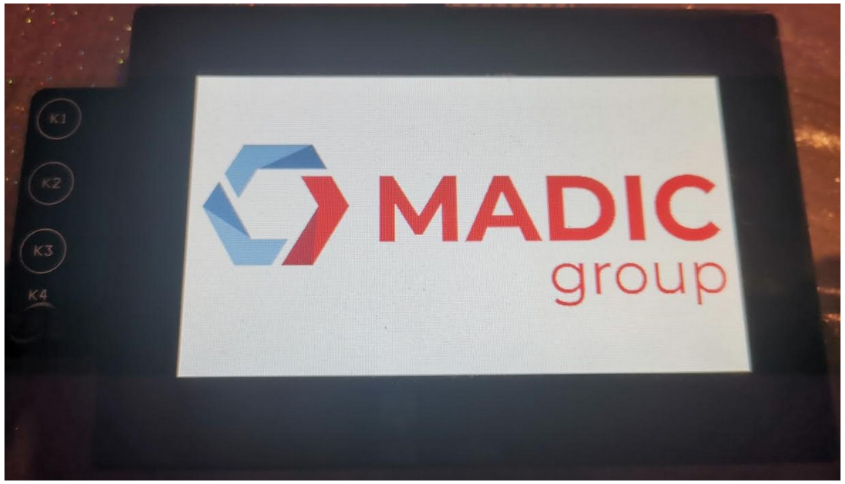
U-Boot : « *BMP* »

- After this trivial modification, we are able to play with our bmp file (without the Vendor Logo) :

```
=> fatload mmc 0:1 0x27000000 madic.bmp
131638 bytes read in 26 ms (4.8 MiB/s)
```

```
=> bmp info 0x27000000
Image size    : 480 x 272
Bits per pixel: 8
Compression   : 0
```

```
=> bmp display 0x27000000
```



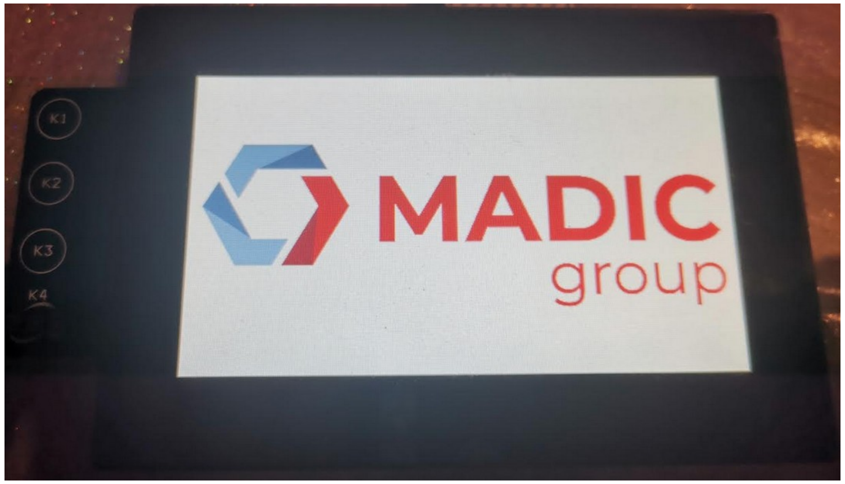
U-Boot : « *BMP* »

- After this trivial modification, we are able to play with our bmp file (without the *Vendor Logo*) :

```
=> fatload mmc 0:1 0x27000000 madic.bmp
131638 bytes read in 26 ms (4.8 MiB/s)
```

```
=> bmp info 0x27000000
Image size     : 480 x 272
Bits per pixel: 8
Compression    : 0
```

```
=> bmp display 0x27000000
```



- For simplicity, we can create the **`loadbmp`** variable (a « script ») that contains the following commands :

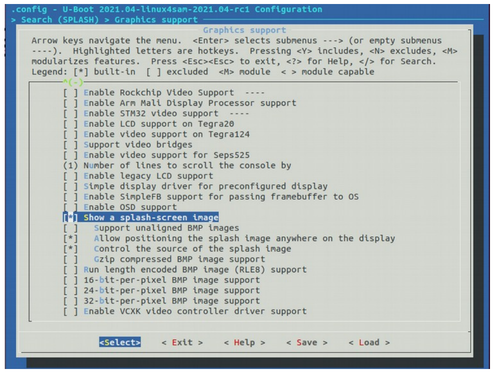
```
=>  env set loadbmp ‘fatload mmc 0:1 0x27000000 madic.bmp && bmp d 0x27000000’
```

U-Boot : ‘Splash Screen’

- Instead of manipulating the *bmp* file from a custom script, U-Boot supports the « **Splash screen** » feature out of the box.

- To use this feature, we can keep our configuration as is with the following addition :

  - `CONFIG_SPLASHSCREEN` (Kconfig)



```
.config - U-Boot 2021.04-linux4sam-2021.04-rc1 Configuration
> Search (SPLASH) > Graphics support
                              Graphics support
   Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus
   ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M>
   modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.
   Legend: [*] built-in  [ ] excluded  <M> module  < > module capable
          ^(-)
          [ ] Enable Rockchip Video Support  ----
          [ ] Enable Arm Mali Display Processor support
          [ ] Enable STM32 video support  ----
          [ ] Enable LCD support on Tegra20
          [ ] Enable video support on Tegra124
          [ ] Support video bridges
          [ ] Enable video support for Seps525
          (1) Number of lines to scroll the console by
          [ ] Enable legacy LCD support
          [ ] Simple display driver for preconfigured display
          [ ] Enable SimpleFB support for passing framebuffer to OS
          [ ] Enable OSD support
          [*] Show a splash-screen image
          [ ]   Support unaligned BMP images
          [*]   Allow positioning the splash image anywhere on the display
          [*]   Control the source of the splash image
          [ ]   Gzip compressed BMP image support
          [ ] Run length encoded BMP image (RLE8) support
          [ ] 16-bit-per-pixel BMP image support
          [ ] 24-bit-per-pixel BMP image support
          [ ] 32-bit-per-pixel BMP image support
          [ ] Enable VCXK video controller driver support

            <Select>    < Exit >    < Help >    < Save >    < Load >
```

- If this option is set, U-Boot will check at startup at least **2 variables** :

  - `splashfile` : the splash screen file (our *bmp* file - ***splash.bmp***)
  - `splashimage` : the address where the splash file will be loaded

- There is also 2 others options that we « **need** » to set from « **menuconfig** » :

  - `CONFIG_SPLASH_SCREEN_ALIGN` (Kconfig)

    - to adjust the position of the splash screen on the display (`splashpos` variable)

  - `CONFIG_SPLASH_SOURCE` (Kconfig)

    - to specify the location of the splash screen file: *eMMC*, *SD*, *USB*, *SATA* (`splashsource` variable)

U-Boot : ‘Splash Screen’

- Then, we just have to set proper ***splashscreen*** env values from U-Boot prompt :

```
=> env set splashfile madic.bmp
=> env set splashimage 27000000
=> env set splashsource mmc_fs
=> env save
Saving Environment to FAT... OK
```

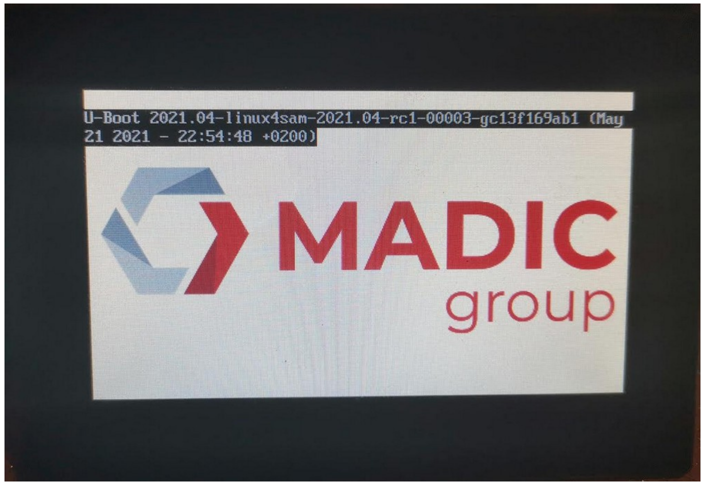
- Our splash file is already centered, but let’s define the **`splashpos`** variable with the following value :

```
=> env set splashpos m,m
```

- « ***m,m*** » for « ***middle,middle*** »

- As already said before, our *splash file* is centered, but the default position could be : x=0,y=0

- In this case, if the **`splashpos`** is as default (0,0), the output banner (bootloader string) is shown, see :



```
U-Boot 2021.04-linux4sam-2021.04-rc1-00003-gc13f169ab1 (May
21 2021 - 22:54:48 +0200)
```

  → *https://source.denx.de/u-boot/u-boot/-/blob/master/common/splash.c#L174*

U-Boot : bootloader string

- To remove the « *bootloader string* »
- Controlled by **`CONFIG_HIDE_LOGO_VERSION`** option
- Not accessible from **`KCONFIG`** :

```
$: git --no-pager grep CONFIG_HIDE_LOGO_VERSION
common/splash.c:#if defined(CONFIG_DM_VIDEO) && !defined(CONFIG_HIDE_LOGO_VERSION)
common/splash.c:#endif /* CONFIG_DM_VIDEO && !CONFIG_HIDE_LOGO_VERSION */
common/splash.c:#if defined(CONFIG_DM_VIDEO) && !defined(CONFIG_HIDE_LOGO_VERSION)
drivers/video/Kconfig:          CONFIG_HIDE_LOGO_VERSION
drivers/video/cfb_console.c:#ifndef CONFIG_HIDE_LOGO_VERSION
include/configs/ge_b1x5v2.h:#define CONFIG_HIDE_LOGO_VERSION
include/configs/ge_bx50v3.h:#define CONFIG_HIDE_LOGO_VERSION
include/configs/gw_ventana.h:#define CONFIG_HIDE_LOGO_VERSION  /* Custom config to hide U-boot version */
scripts/config_whitelist.txt:CONFIG_HIDE_LOGO_VERSION
```

- Must be defined in config board :

```
diff --git a/include/configs/sama5d27_som1_ek.h b/include/configs/sama5d27_som1_ek.h
index 8942d15934..1c61d2b410 100644
--- a/include/configs/sama5d27_som1_ek.h
+++ b/include/configs/sama5d27_som1_ek.h
@@ -36,6 +36,8 @@
                          "bootz 0x22000000 - 0x21000000"
 #endif

+#define CONFIG_HIDE_LOGO_VERSION
+
 /* SPL */
 #define CONFIG_SPL_MAX_SIZE           0x10000
 #define CONFIG_SPL_BSS_START_ADDR     0x20000000
```

MADIC group

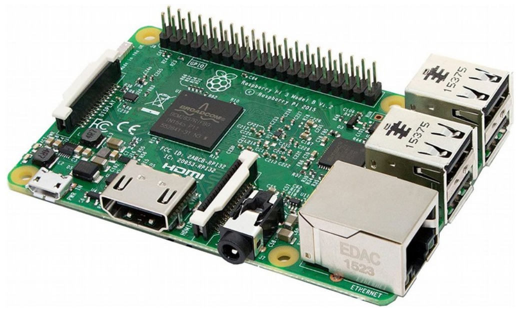
U-Boot : Splash Screen

That’s it

Linux Kernel

Linux Kernel : Setup

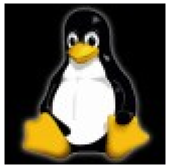
- ***Raspberry-pi 3*** + Official 7" Touchscreen for Raspberry Pi



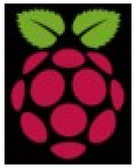
- Image generated with Yocto/OE :
  - Release : dunfell
  - Linux version « **5.4.72** »

Linux Kernel

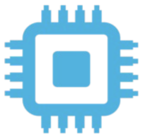
- The Linux kernel has also a static splash-screen support
  - ***Device Drivers → Graphics support → Bootup logo***
- The most famous is « Tux » :

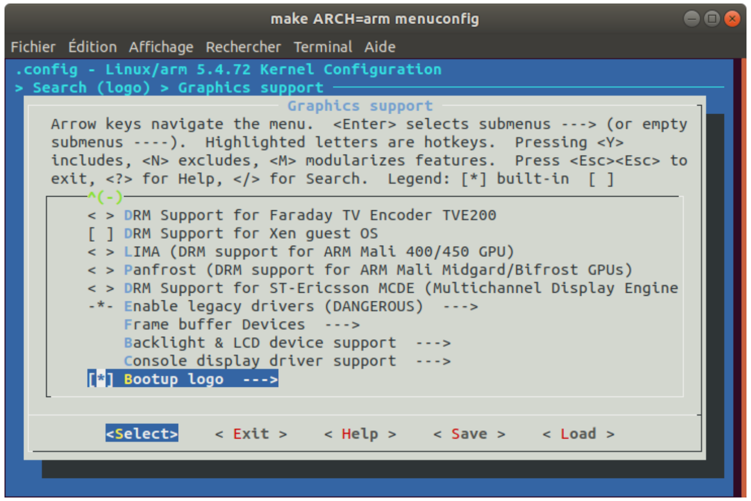


- Overriden for the raspberry-pi (*linux-raspberrypi*) :



- But we will use (another) custom one :





```
make ARCH=arm menuconfig
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
.config - Linux/arm 5.4.72 Kernel Configuration
> Search (logo) > Graphics support
                         Graphics support
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>
  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to
  exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]
        ^(-)
        < > DRM Support for Faraday TV Encoder TVE200
        [ ] DRM Support for Xen guest OS
        < > LIMA (DRM support for ARM Mali 400/450 GPU)
        < > Panfrost (DRM support for ARM Mali Midgard/Bifrost GPUs)
        < > DRM Support for ST-Ericsson MCDE (Multichannel Display Engine
        -*- Enable legacy drivers (DANGEROUS)  --->
            Frame buffer Devices  --->
            Backlight & LCD device support  --->
            Console display driver support  --->
        [*] Bootup logo  --->

        <Select>    < Exit >    < Help >    < Save >    < Load >
```

Linux Kernel

- Instead of using the default logo file « *logo_linux_clut224.ppm* », we'll use a custom one !

- First, let's create a new image that fitts the size of our display (*800x480*)
  - Named « ***logo_lee_linux.png*** » for instance

- Convert it to a **224 colors PPM** formatted image :

```
$: pngtopnm logo_lee_linux.png | ppmquant 224 | pnmnoraw > logo_lee_clut224.ppm
```

- Then, put this one in the right place :

```
$: cp logo_lee_clut224.ppm <linux-tree>/drivers/video/logo
```

- Now we have to proceed with the integration to Linux's sources :

```
diff --git a/drivers/video/logo/Kconfig b/drivers/video/logo/Kconfig
index 6d6f8c08792d..bcda36f8d42d 100644
--- a/drivers/video/logo/Kconfig
+++ b/drivers/video/logo/Kconfig
@@ -28,6 +28,10 @@ config LOGO_LINUX_CLUT224
       bool "Standard 224-color Linux logo"
       default y

+config LOGO_LEE_CLUT224
+        bool "Standard 224-color LEE logo"
+        default y
+
```

Kconfig entry

Linux Kernel

```
diff --git a/drivers/video/logo/Makefile b/drivers/video/logo/Makefile
index 16f60c1e1766..ee3f1c4c6da9 100644
--- a/drivers/video/logo/Makefile
+++ b/drivers/video/logo/Makefile
@@ -5,6 +5,7 @@ obj-$(CONFIG_LOGO)                  += logo.o
...
+obj-$(CONFIG_LOGO_LEE_CLUT224)   += logo_lee_clut224.o
```

Makefile entry

Linux Kernel

```
diff --git a/drivers/video/logo/Makefile b/drivers/video/logo/Makefile
index 16f60c1e1766..ee3f1c4c6da9 100644
--- a/drivers/video/logo/Makefile
+++ b/drivers/video/logo/Makefile
@@ -5,6 +5,7 @@ obj-$(CONFIG_LOGO)                  += logo.o
...
+obj-$(CONFIG_LOGO_LEE_CLUT224)   += logo_lee_clut224.o
```

Makefile entry

```
diff --git a/drivers/video/logo/logo.c b/drivers/video/logo/logo.c
index 141f15a9a459..17b922c8a5e9 100644
--- a/drivers/video/logo/logo.c
+++ b/drivers/video/logo/logo.c
@@ -75,6 +75,10 @@ const struct linux_logo * __ref fb_find_logo(int depth)
             /* Generic Linux logo */
             logo = &logo_linux_clut224;
 #endif
+#ifdef CONFIG_LOGO_LEE_CLUT224
+            /* LEE Linux logo */
+            logo = &logo_lee_clut224;
+#endif
```

add support for LEE/Linux logo for framebuffer console

MADIC group

Linux Kernel

```
diff --git a/drivers/video/logo/Makefile b/drivers/video/logo/Makefile
index 16f60c1e1766..ee3f1c4c6da9 100644
--- a/drivers/video/logo/Makefile
+++ b/drivers/video/logo/Makefile
@@ -5,6 +5,7 @@ obj-$(CONFIG_LOGO)                   += logo.o
...
+obj-$(CONFIG_LOGO_LEE_CLUT224)   += logo_lee_clut224.o
```

Makefile entry

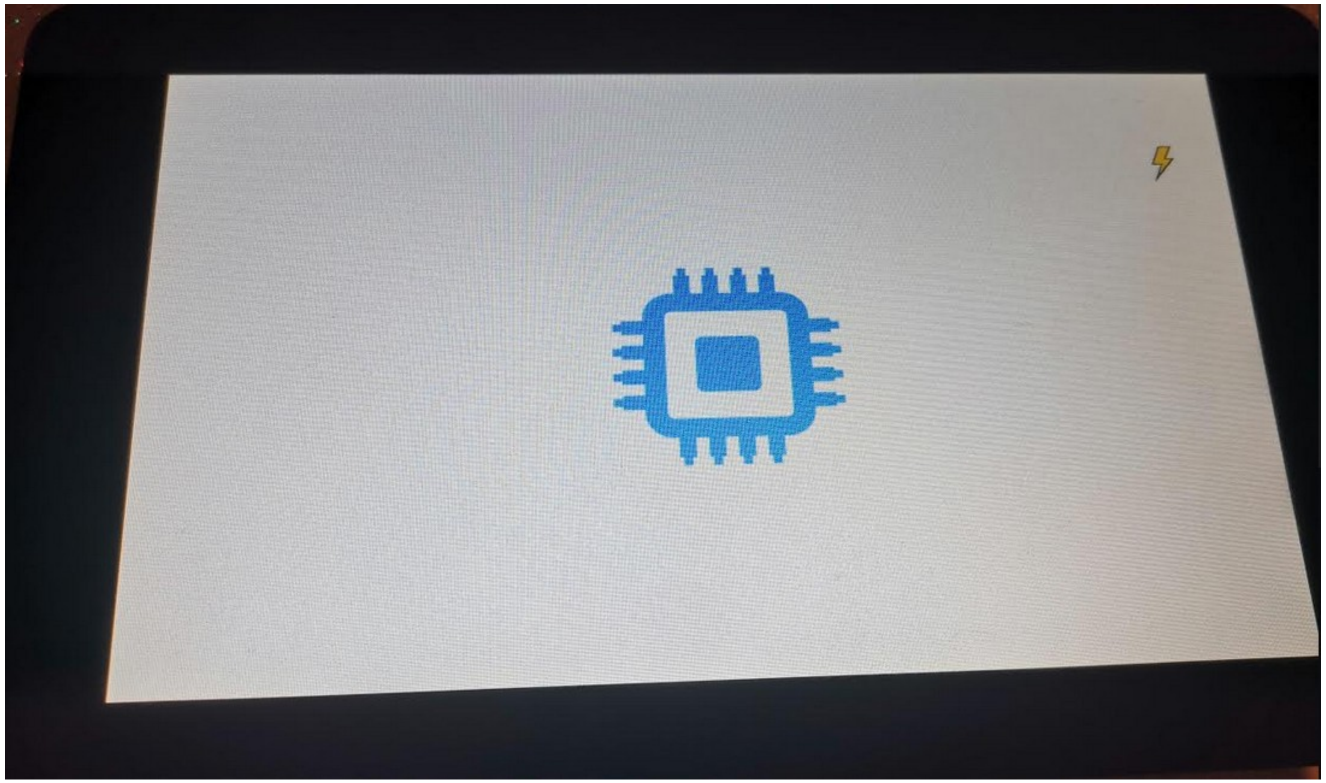
```
diff --git a/drivers/video/logo/logo.c b/drivers/video/logo/logo.c
index 141f15a9a459..17b922c8a5e9 100644
--- a/drivers/video/logo/logo.c
+++ b/drivers/video/logo/logo.c
@@ -75,6 +75,10 @@ const struct linux_logo * __ref fb_find_logo(int depth)
             /* Generic Linux logo */
             logo = &logo_linux_clut224;
 #endif
+#ifdef CONFIG_LOGO_LEE_CLUT224
+            /* LEE Linux logo */
+            logo = &logo_lee_clut224;
+#endif
```

add support for LEE/Linux logo for framebuffer console

```
diff --git a/include/linux/linux_logo.h b/include/linux/linux_logo.h
index d4d5b93efe84..fe7976a63c27 100644
--- a/include/linux/linux_logo.h
+++ b/include/linux/linux_logo.h
@@ -36,6 +36,7 @@ struct linux_logo {
...
+extern const struct linux_logo logo_lee_clut224;
```

add support for LEE/Linux logo for framebuffer console

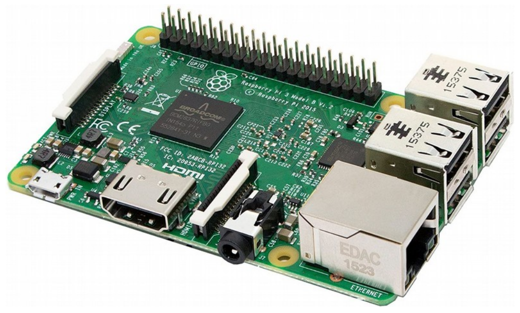
Linux Kernel



Userspace

Userspace : Setup

- ***Raspberry-pi 3*** + Official 7" Touchscreen for Raspberry Pi



- Image generated with Yocto/OE :
  - Release : dunfell
  - Linux version « **5.4.72** »

Userspace : static image

- From Userspace area, it is quite simple to interact with the framebuffer (`/dev/fbX`).
  - ➔ There are many open source projects …
- **fbi**: (from fbida)
  - ➔ https://github.com/fcarlier/fbida

```
root@raspberrypi3:~# fbi -d /dev/fb0 -vt 1 -a logo.png --noverbose
```

- **fbvis**:
  - ➔ https://repo.or.cz/fbvis.git

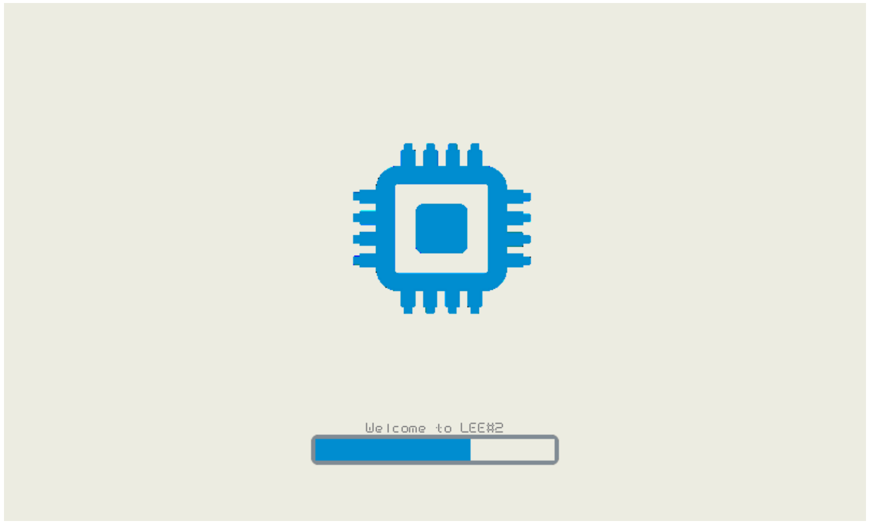
```
root@raspberrypi3:~# echo q | /usr/bin/fbvis logo.png
```

  - ➔ A nice article by Christophe Blaess (in French) talks about fbvis :
    - https://www.blaess.fr/christophe/files/Optimisation-du-temps-de-boot-d-un-systeme-embarque.pdf

Userspace : PSplash

« PSplash is a userspace graphical boot splash screen for mainly embedded Linux devices supporting a 16bpp or 32bpp framebuffer »

- First release in 2006 (OpenHand)
- Git: http://git.yoctoproject.org/cgit/cgit.cgi/psplash/tree/
- Uses **fbdev** graphics to show a logo on a screen during the boot sequence
- The default reference in Yocto/OpenEmbedded
- Supported by Buildroot too
- Fully customizable (progress bar, logo, …)
- External progran can interacts with the main program (FIFO)
  - ➔ Progression and Text - « *psplash-write* »
- Supports 'systemd' (configurable at compile time : --*with-systemd*) from the system bus
- Used by SWUpdate (*swupdate-progress*)



Userspace : PSplash !

- Download it from the official repository :

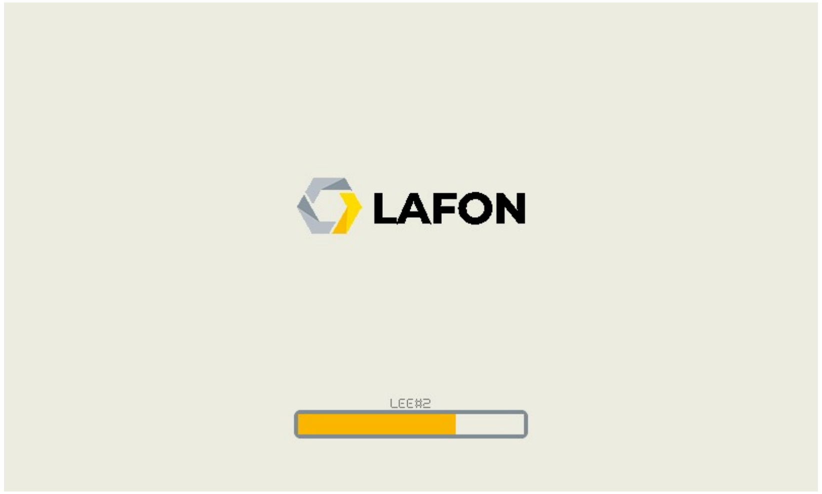
```
$: git clone git://git.yoctoproject.org/psplash
$: cd psplash
```

- Generate the « *custom* » bar file (if any) :

```
$: ./make-image-header.sh ./psplash-custom-bar.png BAR
$: mv ./psplash-custom-bar-img.h psplash-poky-img.h
```

- Generate the « *custom* » logo file :

```
$: ./make-image-header.sh ./psplash-custom-logo.png POKY
$: mv ./psplah-custom-logo-img.h psplash-poky-img.h
```



- Customize colors & options if needed (e.g bar color)

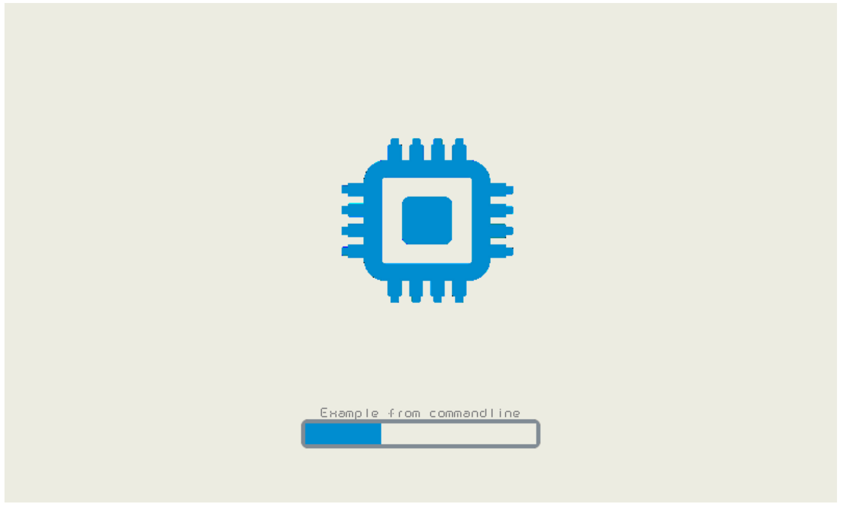
```
diff --git a/psplash-colors.h b/psplash-colors.h
index 82a9893..eb4de9c 100644
--- a/psplash-colors.h
+++ b/psplash-colors.h
@@ -18,7 +18,7 @@
 #define PSPLASH_TEXT_COLOR 0x6d,0x6d,0x70

 /* This is the color of the progress bar indicator */
-#define PSPLASH_BAR_COLOR 0x6d,0x6d,0x70
+#define PSPLASH_BAR_COLOR 0x00,0x8d,0xd0
```

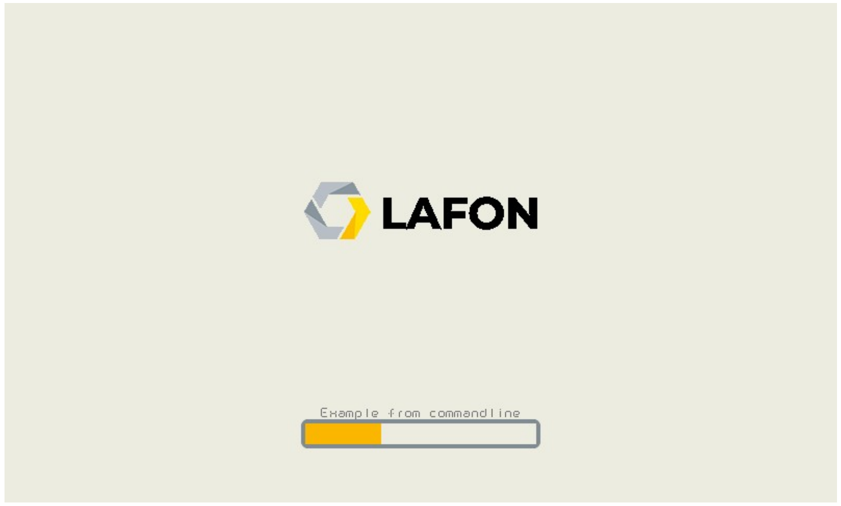
- Many other posibilities : PSPLASH_BACKGROUND_COLOR, PSPLASH_TEXT_COLOR, PSPLASH_STARTUP_MSG, …

Userspace : PSplash !

- Quick test :

```
$: psplash &
$: psplash-write "MSG Example from commandline"
$: psplash-write "PROGRESS 33"
```


Example from commandline



Example from commandline

- To « kill » ***psplash*** properly :

```
$: psplash-write "QUIT"
```

Userspace : EasySplash

« EasySplash is an application that runs early the OS boot for showing graphical animation while the boot process itself happens in the background. »

- First (public) release in 2020
- Git: https://github.com/OSSystems/EasySplash
- *Takes as input zip archives containing a description and PNG-encoded image frames*
- Archive is named **bootanimation.zip** (like for Android)
- Selects a boot animation zipfile from :
  - ***/lib/easysplash/oem/***
  - ***/lib/easysplash/***
- Animation described in « **desc.txt** » file
- « master » now uses **rust** and mp4 as input
- Well integrated with Yocto/OE

yocto PROJECT

MADIC group

Userspace : EasySplash !

- Generate the **bootanimation** (*mp4* -> *png*) => `ffmpeg -i foo.mp4 foo%04d.png`

Userspace : EasySplash !

- Generate the **bootanimation** (*mp4 -> png*)
- Create the « **tree** » with image frames inside :

```
$: tree -L 1 .
.
├── desc.txt
├── part1
├── part2
└── part3
```

Userspace : EasySplash !

- Generate the **bootanimation** (*mp4 -> png*)

- Create the « **tree** » with image frames inside :

```
$: tree -L 1 .
.
├── desc.txt
├── part1
├── part2
└── part3
```

- Create the « **desc.txt** » file :

```
800 480 15
p 1 0 part1
p 1 0 part2
p 1 0 part3
p 1 0 part1
```

  - The first line defines : [width] [weight] [fps]
  - The second : [mode] [num loops] [pause] [name of part directory]

Userspace : EasySplash !

- Generate the **bootanimation** (*mp4 -> png*)
- Create the « **tree** » with image frames inside :

```
$: tree -L 1 .
.
├── desc.txt
├── part1
├── part2
└── part3
```

- Create the « **desc.txt** » file :

```
800 480 15
p 1 0 part1
p 1 0 part2
p 1 0 part3
p 1 0 part1
```

  - The first line defines : [width] [weight] [fps]
  - The second : [mode] [num loops] [pause] [name of part directory]

part 1

part 2

part 3

Userspace : EasySplash !

- Generate the **bootanimation** (*mp4 -> png*)
- Create the « **tree** » with image frames inside :

```
$: tree -L 1 .
.
├── desc.txt
├── part1
├── part2
└── part3
```

- Create the « **desc.txt** » file :

```
800 480 15
p 1 0 part1
p 1 0 part2
p 1 0 part3
p 1 0 part1
```

  - ➔ The <u>first</u> line defines : [width] [weight] [fps]
  - ➔ The <u>second</u> : [mode] [num loops] [pause] [name of part directory]

- So, this means that :
  - ➔ the **3 animations** must be played completely during the boot sequence
  - ➔ Part1 is played 2 times

part 1

FONDÉ à
BORDEAUX
en 1959
part 2

AUTOMATES
TACTILES
part 3

Userspace : EasySplash !

- Generate the **bootanimation** (*mp4 -> png*)
- Create the « **tree** » with image frames inside :

```
$: tree -L 1 .
.
├── desc.txt
├── part1
├── part2
└── part3
```

- Create the « **desc.txt** » file :

```
800 480 15
p 1 0 part1
p 1 0 part2
p 1 0 part3
p 1 0 part1
```

  - The first line defines : [width] [weight] [fps]
  - The second : [mode] [num loops] [pause] [name of part directory]
- So, this means that :
  - the **3 animations** must be played completely during the boot sequence
  - Part1 is played 2 times
- Then, let’s the generate the proper zip file.

```
$: zip -r0 bootanimation.zip desc.txt part1 part2 part3
```

MADIC group

Userspace : EasySplash !

- Download it from the official repository :

```
$: git clone -b 1.0.x https://github.com/OSSystems/EasySplash.git
$: cd EasySplash
```

- Prepare the build for the raspberrypi :

```
$: mkdir build && cd build
$: cmake .. -DDISPLAY_TYPE_GLES=1 -DEGL_PLATFORM_RPI_DISPMANX=1
```

- Run make !

```
$: make
```

- Quick test on the target (oem) :

```
$: root@raspberrypi3:~# easysplash&
root@raspberrypi3:~# [     0.011] [info] [...]   Broadcom Display manager service EGL platform initialized,
using EGL 1.4
[     0.031] [info] [...]   loading animation from zip archive /lib/easysplash/oem/bootanimation.zip
```

Userspace : EasySplash !

- Download it from the official repository :

```
$: git clone -b 1.0.x https://github.com/OSSystems/EasySplash.git
$: cd EasySplash
```

- Prepare the build for the raspberrypi :

```
$: mkdir build && cd build
$: cmake .. -DDISPLAY_TYPE_GLES=1 -DEGL_PLATFORM_RPI_DISPMANX=1
```

- Run make !

```
$: make
```

- Quick test on the target (oem) :

```
$: root@raspberrypi3:~# easysplash&
root@raspberrypi3:~# [     0.011] [info] [...]   Broadcom Display manager service EGL platform initialized,
using EGL 1.4
[     0.031] [info] [...]   loading animation from zip archive /lib/easysplash/oem/bootanimation.zip
```

```
root@raspberrypi3:~# easysplashctl 100 --wait-until-finished
Easysplash PID: 8566
100% reached, will wait until easysplash is finished
EasySplash process terminated successfully
```

Userspace : EasySplash !

Live demo

Userspace : and ?

- **Fbsplash (busybox) :**
  - https://git.buildroot.net/busybox/tree/miscutils/fbsplash.c
- **Bannerd :**
  - https://github.com/alukichev/bannerd
- **Plymouth** :
  - https://github.com/freedesktop/plymouth
- **Dietsplash** :
  - https://github.com/lucasdemarchi/dietsplash
- **PSplash-tn** : A fork of PSplash with GIF support :
  - https://github.com/tano-systems/psplash-tn

Summary

Summary

- Integrating a « Splash Screen » if often a nice to have
- « Splash Screen » integration depends on your needs
- Requires some effort if at bootloader level
- Mostly used in Userspace area (dynamic) - *Android*
- Materials (Yocto/OE) : https://github.com/texierp/meta-splash

Q&A

@texierp

@pjtexier

MADIC group