Add a self-starting program

I. Create a new application named new_program

1. Choose a suitable path

```
[mtk13451@mszsdclx1021 aud-base]$pwd
/proj/mtk13451/codebase/yocto_p1v2/src/apps/aud-base
```

my path: yocto_p1v2/src/apps/aud-base

2. Create a folder to place the source code for new program

```
[mtk13451@mszsdclx1021 aud-base]$mkdir new_program
```

3. Enter the source code folder

```
[mtk13451@mszsdclx1021 aud-base]$cd new_program/
```

4. Write code for new_program

```
[mtk13451@mszsdclx1021 new_program]$vim main.c
```

The main.c looks like this:

```
1 #include <stdio.h>
2
3 int main(void) {
4     printf("<mediatek>This is a new program!\n");
5     return 0;
6 }
```

5. Write Makefile

```
[mtk13451@mszsdclx1021 new_program]$vim Makefile
```

The Makefile looks like this:

```
1  TARGET := new_program
2
3  SOURCES := main.c
4
5  OBJS := $(patsubst %.c,%.o,$(SOURCES))
6
7  all:$(TARGET)
8
9  $(TARGET):$(OBJS)
10     $(CC) -o $@ $^
11
12 $(OBJS):%.o:%.c
13     $(CC) -o $@ -c $<
14
15 clean:
16     rm -rf $(TARGET) $(OBJS)
17
```

II. Add new program to project compilation

1. Go to the suitable rule storage path

```
[mtk13451@mszsdclx1021 recipes-apps]$pwd
/proj/mtk13451/codebase/yocto_p1v2/meta/meta-mediatek-aud/recipes-apps
```

rule path: yocto_p1v2/meta/meta-mediatek-aud/recipes-apps

2. Create a rule storage folder

```
[mtk13451@mszsdclx1021 recipes-apps]$mkdir new-program
```

Created a folder with the name new-program

3. Enter the rule folder

```
[mtk13451@mszsdclx1021 recipes-apps]$cd new-program/
```

4. Create a new rule to call the new program's Makefile

```
[mtk13451@mszsdclx1021 new-program]$vim new-program.bb
```

new-program.bb is new program's rule file.

PS:It should be noted that the new-program.bb is "-" instead of "_".

The new-program.bb looks like this:

```
1  DESCRIPTION = "new_program"
2  LICENSE = "GPLv2"
3
4  APPS_SRC = "${TOPDIR}/../src/apps/aud-base/new_program"
5
6  inherit workonsrc
7
8  WORKONSRC = "${APPS_SRC}"
9
10 do_compile() {
11     make
12 }
13
14 do_install() {
15     install -d ${D}${bindir}
16
17     install -m 755 ${S}/new_program ${D}${bindir}
18 }
19
20 FILES_${PN} += ""
21 INSANE_SKIP_${PN} += "already-stripped ldflags"
22 FILES_${PN}-dev = ""
```

1) ${D} means that when new_program compiled, the path of some files need to be collected into the image .
2) ${bindir} means that usr/bin.
3) ${S} means that the path that new_program source code will store when compiled.
4) install -d means that when new_program is compiled, create a folder that will be collected into the image.
5) install -m means that the new_program executable program under the ${S} will be install to ${D}/usr/bin,and it will be collected into the image.

5. Go to the path of the project compilation rules

```
[mtk13451@mszsdclx1021 images]$pwd
/proj/mtk13451/codebase/yocto_p1v2/meta/meta-mediatek-aud/recipes-audio/images
```

project compilation path: yocto_p1v2/meta/meta-mediatek-aud/recipes-audio/images

6. Add new-program.bb to the project compiled rules

```
[mtk13451@mszsdclx1021 images]$vim mtk-image-aud-8516.bb
```

modify mtk-image-aud-8516.bb like this:

```
20 IMAGE_INSTALL_append = " \
21     ${@base_contains('LICENSE_FLAGS_WHITELIST', 'commercial', 'ffmpeg', '' ,d)} \
22     mdns\
23     openssl \
24     hostapd \
25     wpa-supplicant \
26     alsa-utils \
27     alsa-lib \
28     dhcp-server-config \
29     wpa-supplicant-passphrase \
30     mtkcombotool\
31     mtkwlan \
32     mtkcombo \
33     hostapd \
34     bluetooth \
35     fuse \
36     curl \
37     dhcpcd \
38     appmainprog \
39     AssistantCenter \
40     new-program \
41     ppc \
42     ppccli \
```

add “new-program” at “IMAGE_INSTALL_append”

III. Add self-start features

1. Enter the new program source code path

```
[mtk13451@mszsdclx1021 new_program]$pwd
/proj/mtk13451/codebase/yocto_p1v2/src/apps/aud-base/new_program
```

2. Create a new_program.service file

```
[mtk13451@mszsdclx1021 new_program]$vim new_program.service
```

new_program.service looks like this:

```
1  [Unit]
2  Description=new_program
3
4  [Service]
5  ExecStart=/usr/bin/new_program >ttyS0
6  Type=simple
7  StandardOutput=tty
8  StandardError=tty
9
10 [Install]
11 Alias=new_program
12 WantedBy=multi-user.target
```

“>ttyS0”/”StandardOutput = tty”/“StandardError = tty” is to redirect the output of new_program to the serial port.

3. Enter the new program rule file path

```
[mtk13451@mszsdc1x1021 new-program]$pwd
/proj/mtk13451/codebase/yocto_p1v2/meta/meta-mediatek-aud/recipes-apps/new-program
```

4. modify new_program.bb file

```
1  DESCRIPTION = "new_program"
2  LICENSE = "GPLv2"
3
4  APPS_SRC = "${TOPDIR}/../src/apps/aud-base/new_program"
5
6  inherit workonsrc systemd
7
8  WORKONSRC = "${APPS_SRC}"
9
10 SYSTEMD_PACKAGES = "${PN}"
11 SYSTEMD_SERVICE_${PN} = "new_program.service"
12
13 do_compile() {
14     make
15 }
16
17 do_install() {
18     install -d ${D}${bindir}
19     install -d ${D}${systemd_unitdir}/system
20
21     install -m 755 ${S}/new_program ${D}${bindir}
22     install -m 755 ${S}/new_program.service ${D}${systemd_unitdir}/system
23 }
24
25 FILES_${PN} += "${systemd_unitdir}/system/new_program.service"
26 INSANE_SKIP_${PN} += "already-stripped ldflags"
27 FILES_${PN}-dev = ""
```

The modification is shown in Figure.

IV. Verification

Compile the project, download the image to platform, if see this two serial log:

1. [OK] Started new_program.
2. <mediatek>This is a new program!

means that new program has been self-start success!