Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

Petalinux Auto login and Auto boot

이름 : 김시윤
강사 :이상훈

1.

앞에 문서와 같이 하드웨어를 설계한 후 페타리눅스에 업로드하여 소프트웨어 프로젝트를 진행할 때,
petalinux-create -t apps -n <User App name> –enable 을 하게되면 User 영역의 코드를 작성할 수 있는 공간이 생긴다.
c 파일에는 user code 를 작성하고, 나머지 config 할 설정은
Makefile 을 통해 해주면 된다.

Makefile Auto login 과 Auto boot 는 아래 데이터 시트를 참고한다.

Application Auto Run at Startup

This section explains how to add applications that run automatically at system startup.

Prerequisites

This section assumes that the following prerequisites have been satisfied:

- You have PetaLinux Tools software platform ready for building a Linux system customized to your hardware platform. Please refer to section Import Hardware Configuration for more information.

Steps for Application Auto Run at Startup

If you have prebuilt/newly created custom user application `mystartup` located in your PetaLinux project at `<plnx-proj-root>/components/apps/`, you may want to execute it at system startup. The steps to enable that are:

TIP:

- *If you have prebuilt application and you have not included in PetaLinux Root file system, please refer to Including Prebuilt Applications.*
- *If you want to create custom application and install it in PetaLinux Root file system, please refer to Adding Custom Applications.*

---

1. Change to the application directory.

```
$ cd <plnx-proj-root>/components/apps/mystartup
```

2. Change the `install:` section of the Makefile to copy `mystartup` app to `/etc/init.d/` and create a symbolic link to `/etc/rc5.d/` as follows. These changes will make sure that mystartup app will execute at system startup.

```
$(TARGETINST) -d -p 0755 mystartup /etc/init.d/mystartup
$(TARGETINST) -s /etc/init.d/mystartup /etc/rc5.d/S99mystartup
```

PetaLinux Auto Login

This section explains how to login directly from boot without having to enter login credentials.

Prerequisites

This section assumes that the following prerequisites have been satisfied:

- You have PetaLinux Tools software platform ready for building a Linux system customized to your hardware platform. Please refer to section Import Hardware Configuration for more information.

Steps for PetaLinux Auto Login

1. Create an application called `autologin` using the following command.

```
$ petalinux-create -t apps --name autologin --enable
```

2. Change to the newly created `autologin` application directory.

```
$ cd <plnx-proj-root>/components/apps/autologin
```

4. Modify the Makefile as follows.

   - Change the `install:` section of the Makefile to copy autologin app to `/etc/init.d/` and create a symbolic link to `/etc/rc5.d/` as follows. These changes will ensure that autologin app will execute at system startup.

```
$(TARGETINST) -d -p 0755 autologin /etc/init.d/autologin
$(TARGETINST) -s /etc/init.d/autologin /etc/rc5.d/S99autologin
```

MPU9250 프로젝트에 적용한 예제

```
siyun@siyun-CR62-6M:~$ cd vivado_workspace/mpu9250_controller/software/component
s/apps/device_driver/
```

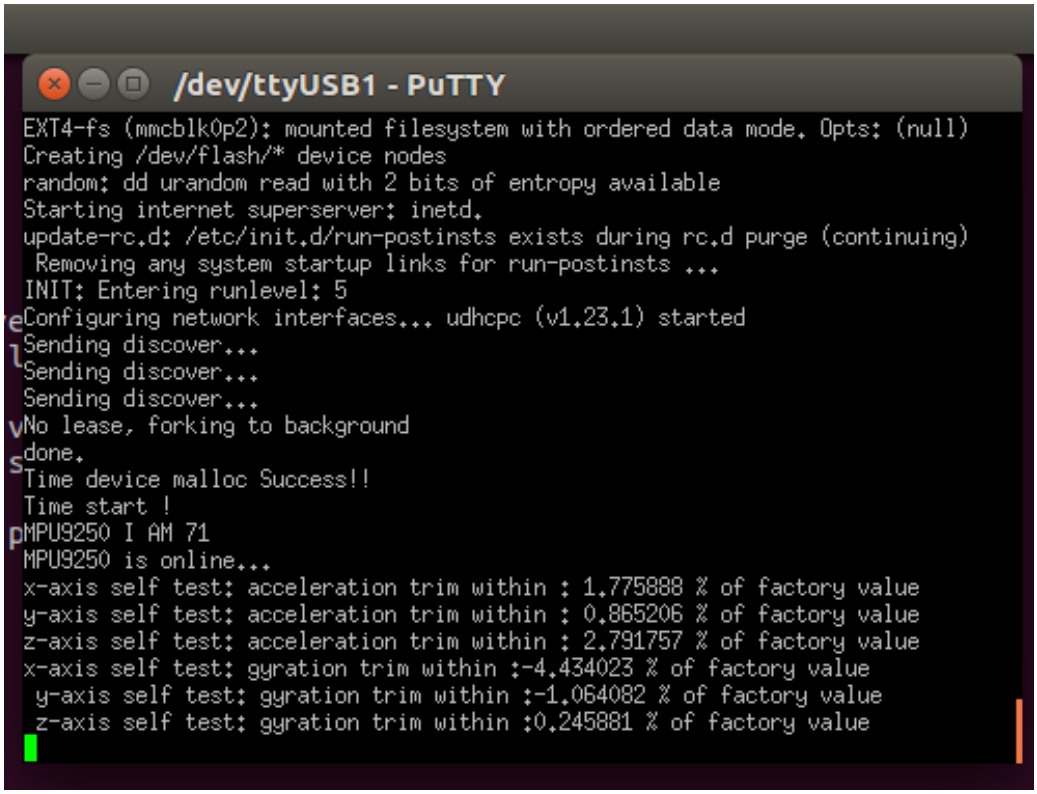
```
siyun@siyun-CR62-6M:~/vivado_workspace/mpu9250_controller/software/components/ap
ps/device_driver$ ls
device_driver.c  Kconfig  Makefile  README
siyun@siyun-CR62-6M:~/vivado_workspace/mpu9250_controller/software/components/ap
ps/device_driver$ vi Makefile
```

```
ifndef PETALINUX
$(error "Error: PETALINUX environment variable not set.  Change to the root of your PetaLinux install, and source the settings.sh file")
endif

include apps.common.mk

APP = device_driver
LDLIBS += -lm


# Add any other object files to this list below
APP_OBJS = device_driver.o

all: build install

build: $(APP)

$(APP): $(APP_OBJS)
	$(CC) $(LDFLAGS) -o $@ $(APP_OBJS) $(LDLIBS)

clean:
	-rm -f $(APP) *.elf *.gdb *.o

.PHONY: install image

install: $(APP)
	$(TARGETINST) -d $(APP) /bin/$(APP)
	$(TARGETINST) -d -p 0755 device_driver /etc/init.d/device_driver
	$(TARGETINST) -s /etc/init.d/device_driver /etc/rc5.d/S99device_driver

%.o: %.c
	$(CC) -c $(CFLAGS) -o $@ $<

help:
	@echo ""
	@echo "Quick reference for various supported build targets for $(INSTANCE)."
	@echo "----------------------------------------------------"
	@echo "  clean                  clean out build objects"
	@echo "  all                    build $(INSTANCE) and install to rootfs host copy"
	@echo "  build                  build subsystem"
	@echo "  install                install built objects to rootfs host copy"
```

여기서 LDLIBS += 은 math.h 를 사용하기 때문에 -lm 옵션을 추가해 준 것이고,

밑에 install 부분이 Auto run 과 Auto login 이다.

동작사진



원래라면 Sending discover 밑에 login 하라는 명령문이 나왔어야 하는데 자동 로그인이되고 user 코드가 자동으로 동작한다.