



PAVELSİS AVİYONİK

Qt Creator ile Yocto SDK Kullanımı

Temmuz 2023

- Bu doküman [linkteki](#) dokümantasyon kullanılarak hazırlanmıştır.

Yocto SDK sayesinde Qt Creator üzerinden oluşturulmuş arayüzleri kullanabilmek için Yocto Project indirmeye gerek yoktur. Yalnızca Yocto SDK paketinin [buradaki](#) linkten indirilmesi yeterlidir. Github'da bulunan [repositoryde](#) pre-built SDK with Qt5 support seçeneğinde bulunan aşağıdaki komutlar takip edilerek bilgisayara Yocto SDK kurulumu gerçekleştirilir.

1. Verilen linkle SDK with Qt5 support image indirilir. Bunun için diskte en az 12 GB yer olması gerekiyor.
2. İçinde yeterince yer bulunan bir dosya yolu seçilerek burası aşağıdaki komutla PATH_TO_SDK olarak set edilir.

```
export PATH_TO_SDK = /path/to/sdk
```

3. İndirilen SDK dosyası seçilen bu dosya yolunda aşağıdaki komutla extract edilir. Bu aşama, indirilen dosya çok büyük olduğundan dolayı uzun sürecektir.

```
sudo tar -C ${PATH_TO_SDK} -xf /path/to/qt5-build-env.tar.bz2
```

- a. Bu adımda eğer önceki export işlemi doğru yapılmadıysa hata alabilirsiniz. Komutla yapılan işlem, temelde sadece indirilen dosyanın extract edilmesi olduğu için PATH_TO_SDK kullanılmadan da yapılabilir. Dosyayı extract etmek istediğimiz dizine giderek burada terminal açılır. Ardından aşağıdaki komut çalıştırılarak dosya extract edilir.

```
sudo tar -C -xf /path/to/qt5-build-env.tar.bz2
```

- b. Bu şekilde bir kullanım normalde tavsiye edilmez. Eğer export işleminde problem yaşıyorsanız dosyayı tekrar indirerek işlemleri baştan yapmanız daha iyi olacaktır.

4. Aşağıdaki komutlar çalıştırılır.

```
sudo mkdir -p /opt/fsl-imx-xwayland/4.14-sumo
```

```
sudo mount -B ${PATH_TO_SDK}/opt/fsl-imx-xwayland/4.14-sumo /opt/fsl-imx-xwayland/4.14-sumo
```

5. Son olarak aşağıdaki adım kullanılarak environment ortamını set etme scripti aktifleştirilir.

```
. /opt/fsl-imx-swayland/4.14-sumo/environment-setup-aarch64-poky-linux
```

- a. Bu adımların yeni bir terminalde SDK kullanılacağı zaman baştan yapılması gerekir.

- b. Buraya kadar her şey doğru gittiye muhtemelen unset library hatası alacaksınız. Aşağıdaki komutu çalıştırdıktan sonra 5. Adımda verilen komutu çalıştırmayı tekrar deneyin.

```
unset LD_LIBRARY_PATH
```

Yocto SDK başarıyla indirilip kurulduktan sonra artık Linux makinenizde kullanmak için gerekli konfigürasyonları yapmaya başlayabilirsiniz. Bu aşamada genel olarak Qt ortamı set edilir ve SDK kurulumu doğru şekilde gerçekleştirilmiş mi kontrol edilir.

1. Build-essentials indirilir.

```
sudo apt-get install build-essential
```

2. Qt Creator indirilir

```
sudo apt-get install qtcreator
```

3. Yocto SDK kurulumu kontrol edilir

```
[[ -f /opt/fsl-imx-xwayland/4.14-sumo/environment-setup-aarch64-poky-linux ]] && echo "Install" || echo "Not Installed "
```

4. Qt5 support kontrol edilir.

```
[[ -f /opt/fsl-imx-xwayland/4.14-sumo/sysroots/x86_64-pokysdk-linux/usr/bin/qt5/qmake ]] && echo "enabled" || echo "off "
```

Yocto SDK kullanımı için 2 farklı yöntem vardır. QtCreator uygulaması linux makinenizde yüklüyse Qt Creator kullanarak gerekli kit ayarları yapılabilir. Ya da bu uygulama sizde yoksa, yani sadece proje dosyalarınız varsa Qt Creator kullanmadan da Yocto SDK kullanarak derleme işlemi yapabilirsiniz.

Qt Creator Olmadan Yocto SDK Kullanımı

1. Yeni bir terminal açılır ve cross compiling ortamı aşağıdaki komutla set edilir.

```
. /opt/fsl-imx-xwayland/4.14-sumo/environment-setup-aarch64-poky-linux
```

2. Örnek Qt5 projelerinin yer aldığı klasör ve içindeki tüm dosyaların komutu çalıştıran kullanıcıya ve kullanıcının ait olduğu gruplara atama işlemi aşağıdaki komutla gerçekleştirilir.

```
sudo chown ${USER}::${GROUPS} -R /opt/fsl-imx-xwayland/4.14-sumo/sysroots/aarch64-poky-linux/usr/share/qt5/examples
```

- a. Bu aşama dosya ve klasörlerin belirli kullanıcı tarafından düzenlenebilmesi için önemlidir. Eğer Linux makinenizde birden fazla kullanıcı yoksa USER kısmını atlayıp direkt kendi kullanıcı adınızı da yazabilirsiniz.

3. Yocto SDK indirildiğinde örnek olarak verilen projelerden biri olan opengl/hellogl2 projesi açılır.

```
cd /opt/fsl-imx-xwayland/4.14-sumo/sysroots/aarch64-poky-linux/usr/share/qt5/examples/opengl/hellogl2
```

4. Sırayla aşağıdaki komutlar gerçekleştirilir.

```
qmake -config release  
make -j 8
```

5. Oluşturulan binary dosyanın ARM64 ile executable olup olmadığı aşağıdaki komut ile kontrol edilir.

```
file hellogl2 | grep -q aarch64 && echo "good" || echo "bad"
```

- a. Bu komutun sonucu olarak eğer good gelirse oluşturulan dosya doğrudur. Hedef cihaza aktararak koşturulabilir. Eğer sonuç bad olarak gelirse az önce gerçekleştirilen make komutu clean edildikten sonra tüm process tekrar edilir.

Qt Creator ile Yocto SDK Kullanımı

1. Yeni bir terminal açılır ve cross compiling ortamı aşağıdaki komutla set edilir.

```
. /opt/fsl-imx-xwayland/4.14-sumo/environment-setup-aarch64-poky-linux
```
2. Aynı terminalde Qt creator aşağıdaki komutla açılır.

```
Qtcreator
```
3. Qt Creator uygulaması açıldıktan sonra Tools --> Options kısmından Kits seçeneği seçilir.
 - a. Daha önce Qt Creator'da hiç kit oluşturmadıysanız bu kısım biraz yabancı gelebilir. Buradaki temel amaç oluşturulan uygulamanın ne kullanılarak compile edileceği, hangi qt versiyonunu kullanacağı, hangi debugger kullanılarak debug edileceğini belirtmektir. Oluşturulan ya da default olarak belirlenen kit, yeni bir proje oluştururken kit seçme ekranında seçilecektir. Eğer farklı platformlarda derlenebilen bir uygulama geliştirmek isteniyorsa yani cross compiling yapılacaksa burada default çıkan kit değil birazdan yapacağınız gibi kendi oluşturduğunuz kit kullanılmalıdır.
4. Kits ekranında add manual seçilerek yeni oluşturulan kite X-compile-ARM-64bit gibi bir isim verilir. Qt version seçilirken qmake yolu olarak aşağıdaki yol girilir. Apply tuşuna basılarak yeni oluşturulan kite bu qt versiyonu kaydedilir.

```
/opt/fsl-imx-xwayland/4.14-sumo/sysroots/x86_64-pokysdk-linux/usr/bin/qt5/qmake
```

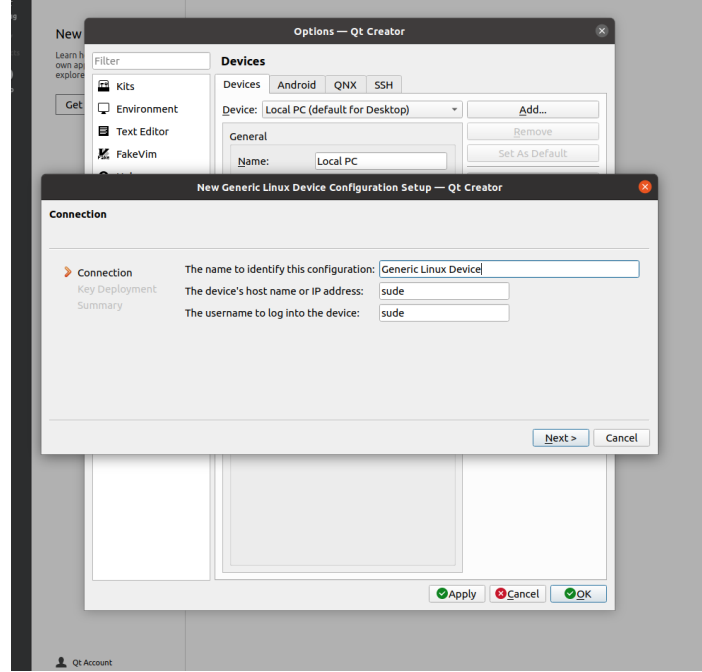
 - a. Bu aşamalarda girdiğiniz yollar kaydedilemediği için kırmızı şekilde görünüyordur. Bunu gidermek için browse seçeneğine tıklayarak bu yolu elle seçmeniz gerekiyor.
5. C compilerı olarak aşağıdaki yol girilir ve Compilers tab olarak ABI arm-linux-generic-elf-64bit seçilir.

```
opt/fsl-imx-xwayland/4.14-sumo/sysroots/x86_64-pokysdk-linux/usr/bin/aarch64-poky-linux/aarch64-poky-linux-gcc
```
6. C++ compilerı olarak aşağıdaki yol girilir ve Compilers tab olarak ABI arm-linux-generic-elf-64bit seçilir.

```
/opt/fsl-imx-xwayland/4.14-sumo/sysroots/x86_64-pokysdk-linux/usr/bin/aarch64-poky-linux/aarch64-poky-linux-g++
```
7. Debugger yolu olarak aşağıdaki yol girilir.

```
/opt/fsl-imx-xwayland/4.14-sumo/sysroots/x86_64-pokysdk-linux/usr/bin/aarch64-poky-linux/aarch64-poky-linux-gdb
```

8. Devices seçeneğinde manage tuşuna basılarak yeni bir device eklenir. Generic Linux Device adında yeni bir linux makinesi eklenir. IP adresi gibi çeşitli özellikler girilerek bu kısım özelleştirilir. Aşağıda bir örneği verilmiştir.



9. Kits kısmına dönülüp device type olarak oluşturulan bu generic linux device seçilir.

10. Sysroot yolu olarak aşağıdaki yol girilir.

/opt/fsl-imx-xwayland/4.14-sumo/sysroots/aarch64-poky-linux

11. Apply tuşuna basılarak tüm özellikler kaydedilir ve Qt Creatordan çıkılır.

En sonda oluşan kit ayarları aşağıdaki gibi görünmelidir.

