
Spis treści

Wstęp	10
1. Yocto	25
1.1. Podstawowe komponenty	26
1.2. Metadane	26
1.3. Przygotowanie systemu	28
1.4. Rozpoczęcie pracy z Yocto	29
1.4.1. Konfiguracja GIT	29
1.4.2. fsl-community-bsp-platform	29
1.4.3. Yocto bez dodatków	31
1.4.4. Wersje Yocto	31
1.4.5. Pliki źródłowe	32
1.4.6. Konfiguracja początkowa	35
1.4.7. Kompilacja	38
1.4.8. Wyniki kompilacji	39
1.4.9. Przygotowanie karty SD	40
1.4.10. Konsola szeregową	41
1.4.11. Uruchomienie systemu	41
1.4.12. SDK	42
1.5. Praca z pakietami	45
1.5.1. Repozytoria	46
1.5.2. Dodatkowe pakiety	48
1.5.3. Przykład: Openvpn	49
1.5.4. Przykład: GStreamer	50
1.5.5. Pomijanie pakietów ze względu na licencję	51
1.5.6. Przykład: OpenCV	51

1.5.7.	Przykład: QT5	52
1.5.8.	Dodawanie własnych pakietów i meta-layerów	55
1.5.9.	Przykład: Smartmontools	57
1.5.10.	Przykład: własna aplikacja	59
1.5.11.	Własna platforma	60
1.6.	Debugowanie aplikacji	64
1.6.1.	Sieć – sposób 1	64
1.6.2.	Sieć – sposób 2 – ifplugd	65
1.6.3.	Metody komunikacji z urządzeniem	67
1.6.4.	Narzędzia wymagane na urządzeniu	69
1.7.	Wybrane funkcje Yocto	71
1.7.1.	Hob.....	71
1.7.2.	Pliki pobierane z Internetu	73
1.7.3.	Sstate-cache	74
1.7.4.	Przydatne komendy bitbake	76
1.7.5.	Szczegółowe statystyki.....	78
1.7.6.	Prelinkowanie.....	78
1.8.	Krytyka Yocto	79
2.	Praca z Eclipse	81
2.1.	Instalacja Eclipse	82
2.1.1.	Pobranie pakietu.....	82
2.1.2.	Java.....	83
2.1.3.	Rozpakowanie i uruchomienie	83
2.2.	Konfiguracja Eclipse do pracy z kodem źródłowym jądra	84
2.2.1.	Tworzenie nowego projektu	84
2.2.2.	Opcje projektu dotyczące indeksowania	85
2.2.3.	Opcje projektu dotyczące kompilacji	91
2.2.4.	Indeksowanie źródeł	92
2.2.5.	Prawidłowo skonfigurowane IDE	93
2.2.6.	Kompilacja jądra	95
2.3.	Moduły jądra rozwijane poza drzewem	96
2.3.1.	Przykładowy moduł.....	97
2.3.2.	Projekt w Eclipse	97
2.4.	Aplikacje przestrzeni użytkownika	99

2.4.1.	Konfiguracja połączenia sieciowego	100
2.4.2.	Projekt w Eclipse	102
2.4.3.	Dodatkowe pliki źródłowe i budowanie	103
2.4.4.	Podłączenie Eclipse do zdalnego systemu.....	104
2.4.5.	Zdalne uruchamianie aplikacji	106
2.4.6.	Zdalne debugowanie	109
2.5.	Debugowanie jądra	110
2.5.1.	KGDB	111
2.5.2.	Użycie KGDB	112
2.5.3.	Debugowanie modułów jądra	113
2.5.4.	Debugowanie jądra za pomocą Eclipse.....	114
3.	Buildroot.....	117
3.1.	Wersje	118
3.2.	Instalacja	119
3.3.	Korzystanie z Buildroota	119
3.4.	Konfiguracja.....	122
3.4.1.	Traget options	122
3.4.2.	Opcje budowania	123
3.4.3.	Toolchain	125
3.4.4.	Kompilacja toolchaina dla i.MX6.....	125
3.4.5.	Zewnętrzny toolchain	129
3.4.6.	Porównanie toolchainów	130
3.4.7.	Konfiguracja systemu	131
3.4.8.	Jądro Linux	133
3.4.9.	Wybór programów i bibliotek.....	134
3.4.10.	Format docelowego systemu plików	137
3.4.11.	Bootloader	138
3.4.12.	Dodatkowe pliki, katalogi i skrypty	140
3.4.13.	Kompilacja.....	142
3.4.14.	Konfiguracja wybranych komponentów	142
3.4.15.	Uruchomienie.....	143
3.5.	Kompilacja zewnętrznych programów	144
3.5.1.	SDK i Toolchain z Buildroota	144
3.5.2.	Integracja z Eclipse	145

3.5.3.	Dodawanie własnych pakietów	145
4.	Timesys LinuxLink	151
4.1.	Obrazy demonstracyjne	153
4.2.	Praca z LinuxLink	155
4.2.1.	Workorder	156
4.2.2.	Konfiguracja BSP	156
4.3.	Posługiwanie się BSP	163
4.3.1.	Instalacja i uruchomienie	164
4.3.2.	Dostęp do plików źródłowych	166
4.3.3.	Factory	167
4.4.	Timestorm IDE	168
5.	Wyświetlacze i grafika	171
5.1.	Przegląd interfejsów graficznych	172
5.1.1.	Proste interfejsy graficzne	172
5.1.2.	Układy podłączane szeregowo	173
5.1.3.	Wyświetlacze podłączane równoległe	174
5.1.4.	Konfiguracja sterowników wyświetlaczy w jądrze	180
5.1.5.	Interfejsy różnicowe	186
5.2.	Bufor ramki i inne interfejsy	188
5.2.1.	X-serwer	188
5.2.2.	fbX	189
5.3.	Konfiguracja wyświetlaczy i.MX6	191
5.3.1.	Parametry jądra	191
5.3.2.	fbset	191
5.3.3.	xrandr	195
5.3.4.	Sprzętowy overlay	196
5.3.5.	Double buffering	198
5.3.6.	Wygaszanie konsoli	198
5.3.7.	Kursor konsoli	198
5.4.	Akceleracja 3D	200
5.4.1.	OpenGL/OpenVG/OpenCL	200
5.4.2.	Vivante	202
5.4.3.	Etnaviv	203
6.	GStreamer i przetwarzanie obrazu	205

6.1.	GStreamer	205
6.1.1.	Wsparcie dla i.MX6	206
6.2.	Elementy strumienia.....	206
6.2.1.	Elementy specyficzne dla i.MX6	208
6.2.2.	Pierwszy pipeline	209
6.2.3.	Odtwarzanie filmu.....	211
6.2.4.	Kontrolowanie pipeline'u	212
6.2.5.	Kamera MIPI	213
6.2.6.	Więcej przykładów strumieni dla i.MX6.....	216
6.2.7.	Odbiór telewizji cyfrowej.....	221
6.3.	Korzystanie z GStreamer w aplikacjach	226
6.3.1.	Serwer RTSP	226
6.3.2.	Integracja z QT.....	229
6.4.	OpenCV	232
6.4.1.	Obraz z kamery i.MX6	235
6.4.2.	Kody kreskowe i QR.....	239
7.	Konfiguracja niskopoziomowa	241
7.1.	Uruchamianie systemu – bootloader	242
7.1.1.	HAB.....	243
7.1.2.	USB Recovery	243
7.1.3.	Start z karty SD	245
7.2.	U-BOOT.....	246
7.2.1.	Kompilacja i uruchomienie.....	246
7.2.2.	Rejestry jednorazowo programowalne	248
7.2.3.	Komendy U-BOOT-a	249
7.2.4.	Zmienne konfiguracyjne	250
7.2.5.	Komenda bdfinfo	252
7.2.6.	Skrypt uruchamiający system	252
7.3.	Alternatywne bootloadery	254
7.4.	Wersje jądra Linux dla Freescale i.MX6	255
7.4.1.	Peryferie i PIN-y procesora	258
7.4.2.	Magistrale i model urządzeń	262
7.4.3.	Linux bez <i>Device Tree</i>	264
7.5.	<i>Device Tree</i>	265

7.5.1.	Budowa drzewa – I2C w Wandboard Solo	267
7.5.2.	Sterownik kontrolera I2C	270
7.5.3.	Dokumentacja w jądrze	272
7.5.4.	Urządzenia I2C – organizacja i komunikacja	272
7.5.5.	Sterowniki I2C – termometr	277
7.5.6.	Sterowniki I2C – ekspander I/O	278
7.5.7.	Termometr 1-Wire	280
7.5.8.	Przykład – uproszczenie zarządzania projektem	285
7.5.9.	Przykład – przekazywanie parametrów jądra	286
7.5.10.	Przykład – izolacja własnego kodu	287
7.5.11.	Przykład – sterowanie zasilaniem peryferii	288
7.6.	Firmware dla urządzeń	290
7.6.1.	Infrastruktura w jądrze	290
7.6.2.	Skąd wziąć firmware	293
7.6.3.	Przykładowy sterownik – Wi-Fi	293
7.6.4.	Przykładowy sterownik – Tuner DVB-T	294
7.7.	Oszczędzanie energii	295
7.7.1.	CPU <i>Frequency Scaling</i>	295
7.7.2.	Tryby uśpienia	299
7.7.3.	Pomiary zużycia energii	302
7.8.	Koprocesor kryptograficzny	303
7.8.1.	Przykład – openssl	304
7.8.2.	Przykład – IPSec w IPv6	307
7.9.	CAN	314
7.9.1.	SocketCAN w Linuksie	315
7.9.2.	Narzędzia przestrzeni użytkownika	317
7.9.3.	Wirtualny kontroler CAN	318
7.9.4.	Fizyczny kontroler CAN	318
7.9.5.	Przykładowy program	319
7.10.	Bluetooth	321
7.11.	Udev	324
7.11.1.	Stała nazwa interfejsu sieciowego	325
7.11.2.	Nazwy urządzeń USB	325
7.11.3.	Nazwy interfejsów sieciowych zależne od portu	329

7.12.	Uruchamianie usług	331
7.12.1.	/etc/inittab.....	331
7.12.2.	/etc/init.d/rcS	331
7.12.3.	Prosty skrypt startowy – /etc/init.d/SXXusługa	332
7.12.4.	start-stop-daemon	332
7.12.5.	Skrypt startowy dla klasycznego inita (Debian)	334
7.12.6.	/etc/rc.local	334
7.12.7.	Uruchamianie usług przez inetd.....	335
7.12.8.	Systemd	336
7.13.	Systemy plików dla urządzeń wbudowanych	341
7.13.1.	Urządzenia blokowe.....	342
7.13.2.	Urządzenia MTD	344
7.13.3.	Urządzenia blokowe oparte na pamięciach Flash	346
7.14.	Szybki start.....	347
7.14.1.	Metodologia pracy	348
7.14.2.	Pomiar czasu	349
7.14.3.	Techniki optymalizacji	350
A.	Dodatek technologiczny	355
A.1.	Technologia montażu układów BGA – PoP (<i>Package on Package</i>)	355
A.2.	Metody montażu układów PoP	357
A.3.	Montaż jednoetapowy typu <i>in-line</i>	358
A.4.	Montaż dwuetapowy typu <i>pre-stacking</i>	362
A.5.	Defekty w montażu układów PoP.....	364
A.6.	Standaryzacja modułów PoP – JEDEC	370
	Bibliografia.....	371
	Spis rysunków	373