

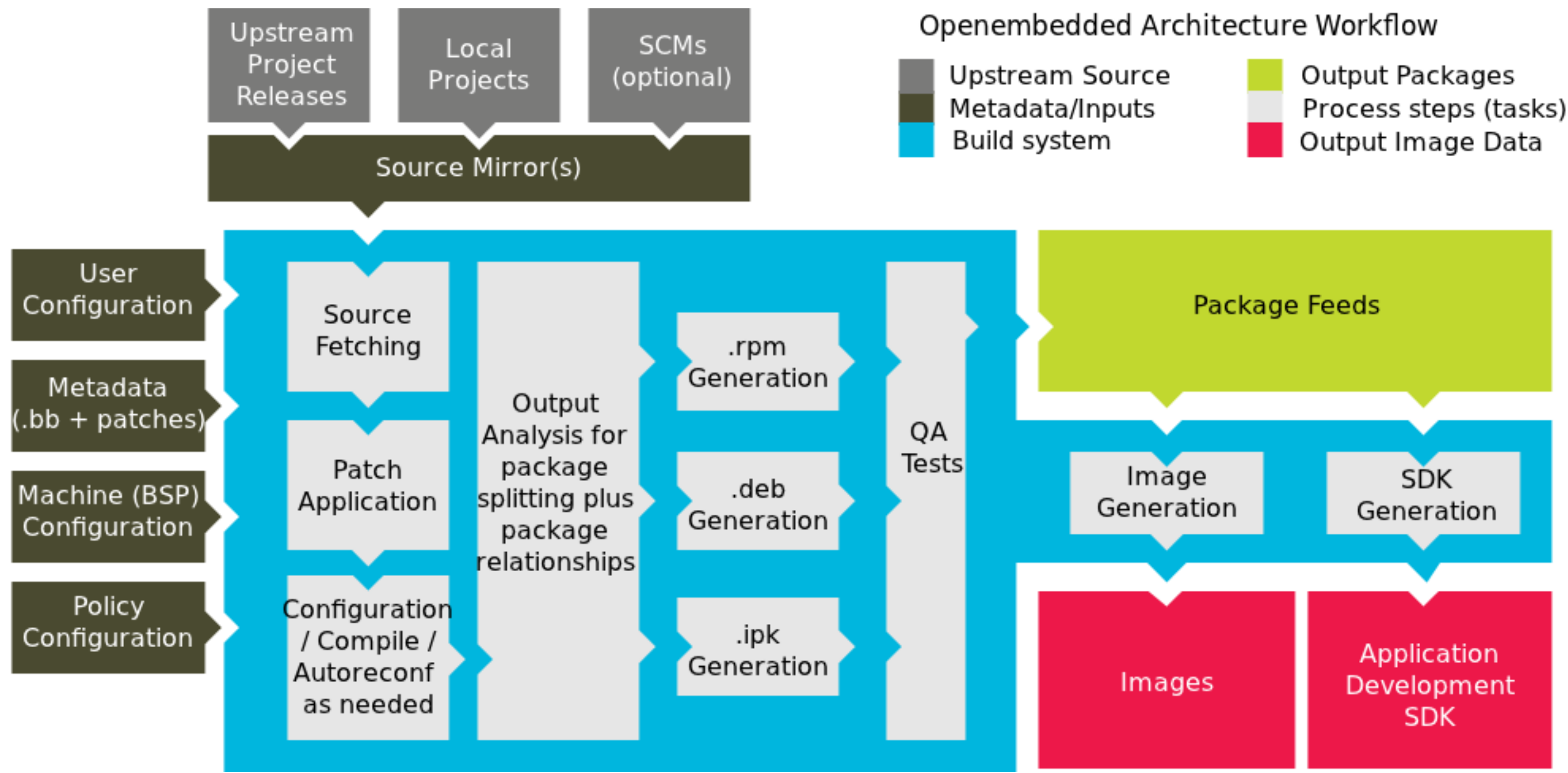
WSB - BUDOWANIE DYSTRYBUCJI RPI ZA
POMOCĄ

YOCTO + BITBAKE

YOCTO PROJECT

- ▶ <https://www.yoctoproject.org/>
- ▶ ogłoszony przez Linux Foundation w 2010, a wystartowany w marcu 2011
- ▶ celem jest usprawnienie procesu tworzenia wbudowanych dystrybucji Linuxa
- ▶ OpenEmbedded - framework automatyzujący budowanie dystrybucji Linuxa dla systemów wbudowanych
- ▶ BitBake - narzędzie budujące
- ▶ Poky - referencyjna dystrybucja Yocto Project

OPENEMBEDDED ARCH YOCTO



YOCTO LAYERS

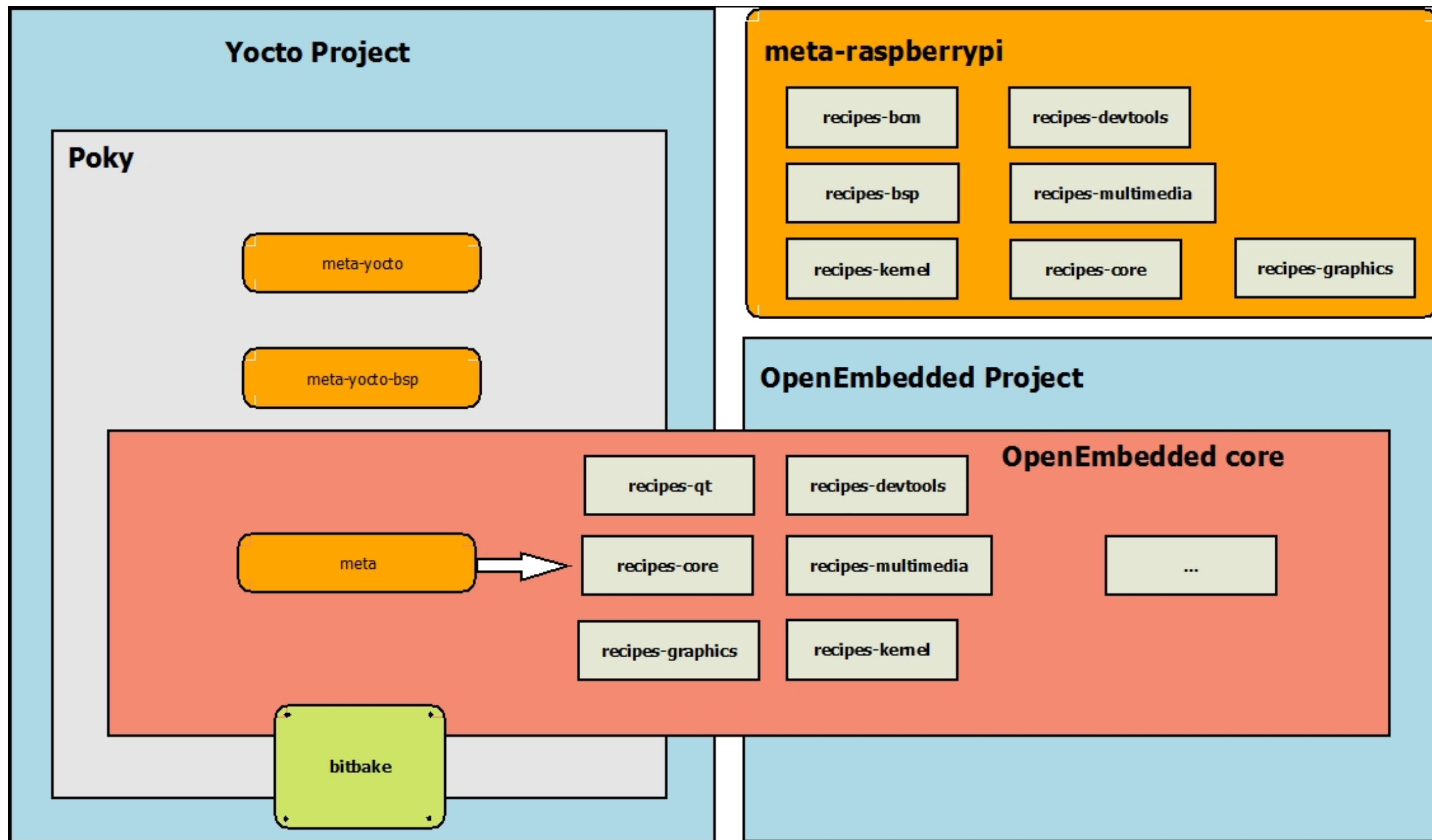
- ▶ repozytoria zawierające metadane mówiące systemowi budowania co robić
 - ▶ zawierają recepty, klasy, pliki konfiguracyjne oraz pliki modyfikujące metadane
- ▶ dostarczane przez inne warstwy
- ▶ •indeks warstw kompatybilnych z Yocto Project:
 - ▶ <https://www.yoctoproject.org/software-overview/layers/>
- ▶ warstwy kompatybilne z OpenEmbedded:
 - ▶ <http://layers.openembedded.org/layerindex/branch/master/layers/>
- ▶ podstawowe warstwy dostarczane w ramach dystrybucji Poky:
 - ▶ meta - rdzeń OpenEmbedded z metadanymi wspierającymi architektury ARM, ARM64, x86, x86-64, PowerPC, MIPS, MIPS64 oraz QEMU
- ▶ meta-poky - metadane dla dystrybucji Poky
- ▶ meta-yocto-bsp - metadane dotyczące sprzętu referencyjnego

Bitbake Layering

Additional Layers	Custom Layer (meta-myproject)
	Software Layer (meta-baryon)
	Kernel Enablement Layer (experimental/meta-x32)
	Hardware Enablement Layer (meta-intel/meta-fsl-ppc)
Default Layers	Yocto Layer ./meta-yocto
	OE-core ./meta

BOARD SUPPORT PACKAGE DLA RPI

- ▶ <http://layers.openembedded.org/layerindex/branch/master/layer/meta-raspberrypi/>
- ▶ **wymaga warstw**
- ▶ • openembedded-core - meta z dystrybucji Poky
 - ▶ <git://git.yoctoproject.org/poky.git>
 - ▶ • meta-oe - <git://git.openembedded.org/meta-openembedded>
- ▶ <https://media.readthedocs.org/pdf/meta-raspberrypi/latest/meta-raspberrypi.pdf>
- ▶ • wspierane maszyny: raspberrypi, raspberrypi0, raspberrypi0-wi, raspberrypi2, raspberrypi3, raspberrypi3-64, raspberrypi-cm, raspberrypi-cm3



DIGI EMBEDDED YOCTO PLATFORM

DIGI YOCTO USER SPACE

DIGI BSP RELEASE

DIGI PLATFORM SUPPORT

META-DIGI

DIGI SOFTWARE EXTENSIONS

DIGI TRUSTFENCE™

CLOUD INTEGRATION

SUPPORTED CLOUDS

DIGI REMOTE
MANAGER*

AMAZON
WEB
SERVICES*

FEATURES

SECURE FIRM-
WARE UPDATE

DEVICE CONFIG

DEVICE
MONITORING

DATA STREAMS
& AGGREGATION

SECURE FIRMWARE UPDATE

CELLULAR MODEM SUPPORT

NETWORK FAILOVER*

DIGI EMBEDDED APIs*

YOCTO PROJECT RELEASE

META-FREESCALE

META-QT5

META-SWUPDATE

NXP BSP RELEASE

GRAPHICAL HARDWARE
ACCELERATION*

VIDEO HARDWARE
ACCELERATION*

CRYPTOGRAPHIC
ACCELERATION*

DIGI LINUX KERNEL

DIGI BSP RELEASE

DIGI PLATFORM SUPPORT

DIGI BSP EXTENSIONS

POWER MANAGEMENT
OPTIMIZATIONS

WIRELESS

DIGI TRUSTFENCE™

MCA™ AND IO EXPANDER*

NXP BSP RELEASE

UPSTREAM KERNEL.ORG
STABILITY PATCHES

DIGI U-BOOT

DIGI BSP RELEASE

DIGI PLATFORM SUPPORT

DIGI BSP EXTENSIONS

DIGI TRUSTFENCE™

USABILITY ENHANCEMENTS

NXP BSP RELEASE

UPSTREAM
DENX RELEASE

PRE-COMPILED IMAGES FOR
ALL SUPPORTED PLATFORMS

DIGI QUALITY APPROVED

ONLINE DOCUMENTATION

DEVELOPMENT TOOLS

PRE-COMPILED SDKs
INCLUDING TOOLCHAINS

BUILD YOUR OWN
CUSTOM SDKs

USE GRAPHICAL IDEs LIKE
ECLIPSE OR QT CREATOR

*ONLY ON SUPPORTED PLATFORMS OR RELEASES

SETUP PROJEKTU

- ▶ Ubuntu 14 LTS
 - ▶ `mkdir -p ~/rpi/sources`
 - ▶ `cd ~/rpi/sources`
 - ▶ Pobranie warstw :
 - ▶ - poky
 - ▶ - meta-openembedded
 - ▶ - meta-raspberrypi

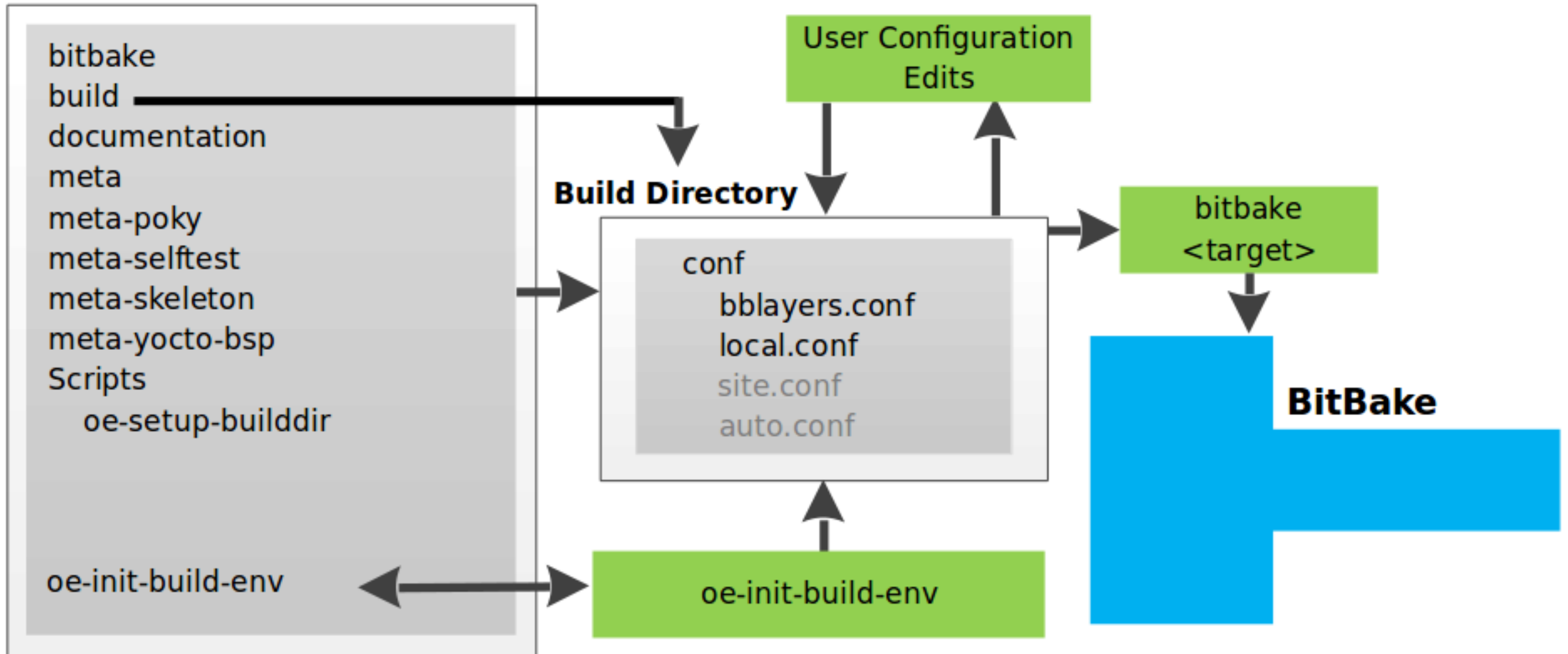
ZRZUT PROJEKTÓW

- ▶ `git clone -b krogoth git://git.yoctoproject.org/poky`
- ▶ `git clone -b krogoth git://git.openembedded.org/meta-openembedded`
- ▶ `git clone -b krogoth git://git.yoctoproject.org/meta-raspberrypi`

LIBKI UBUNTU I DEBIANA

- ▶ `sudo apt-get install gawk wget git-core diffstat unzip
texinfo gcc-multilib build-essential chrpath socat libsdl1.2-
dev xterm`

Source Directory (e.g. poky directory)



Raspberry Pi machines	Model	SOC version
raspberrypi0	Raspberry Pi Zero v1.3	BCM2835
raspberrypi0-wifi	Raspberry Pi Zero Wireless	BCM2835
raspberrypi2	Raspberry Pi 2 (model B)	BCM2836
raspberrypi3-64	Raspberry Pi 3 64 bits (model B)	BCM2837
raspberrypi3	Raspberry Pi 3 (model B)	BCM2837
raspberrypi-cm3	Raspberry Pi Compute module 3	BCM2837
raspberrypi-cm	Raspberry Pi Compute module 1	BCM2835
raspberrypi	?	?

INICJALIZACJA POWŁOKI BUDUJĄCEJ

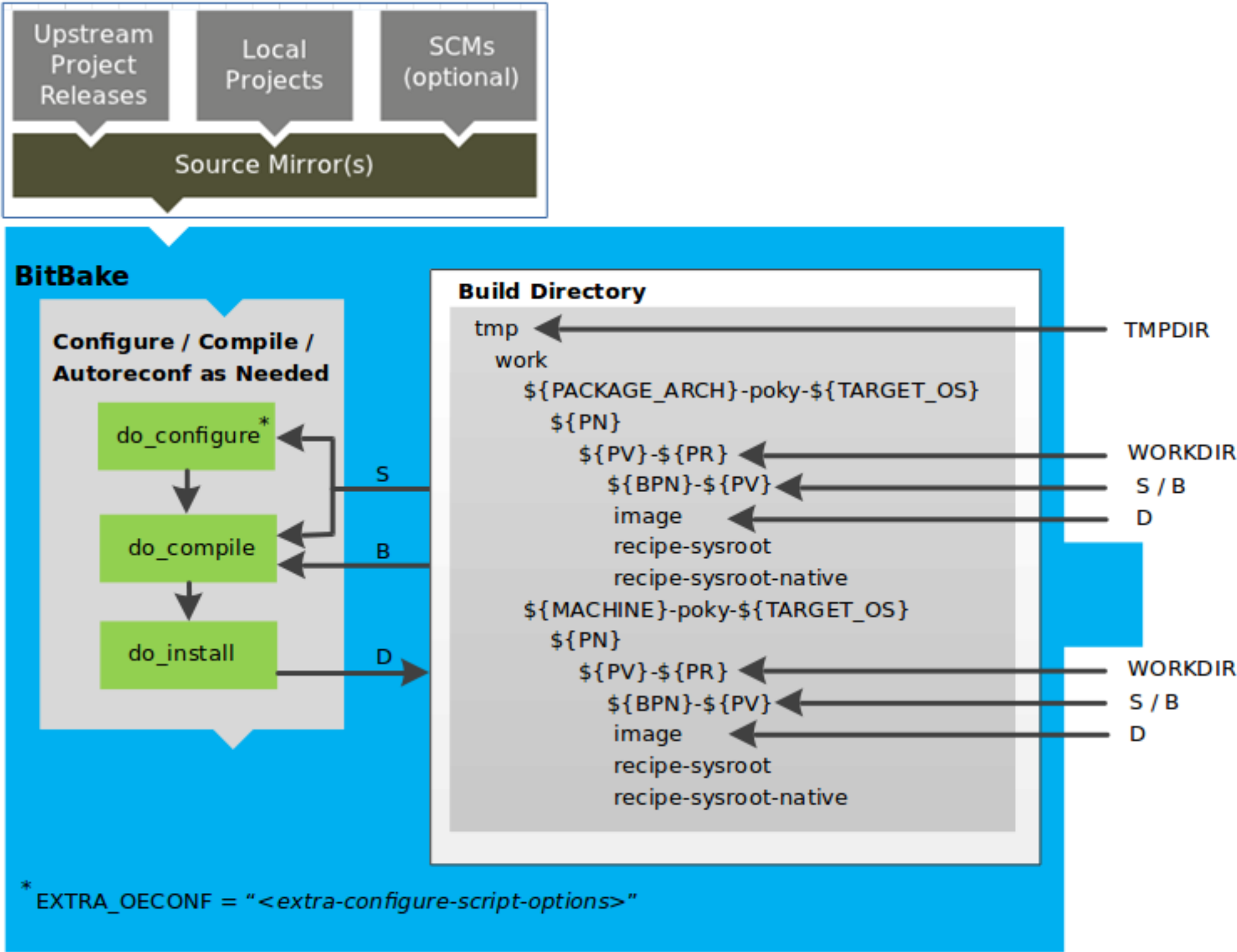
- ▶ `cd ~/rpi/`
- ▶ `source sources/poky/oe-init-build-env rpi-build`

MODYFIKACJA LOCAL.CONF

- ▶ `echo 'MACHINE = "raspberrypi3"' >> conf/local.conf`
- ▶ `echo 'PREFERRED_VERSION_linux-raspberrypi = "4.%"' >> conf/local.conf`
- ▶ `echo 'DISTRO_FEATURES_remove = "x11 wayland"' >> conf/local.conf`
- ▶ `echo 'DISTRO_FEATURES_append = " systemd"' >> conf/local.conf`
- ▶ `echo 'VIRTUAL-RUNTIME_init_manager = "systemd"' >> conf/local.conf`

MODYFIKACJA BBLAYERS.CONF

- ▶ # LAYER_CONF_VERSION is increased each time build/conf/bblayers.conf
- ▶ # changes incompatibly
- ▶ POKY_BBLAYERS_CONF_VERSION = "2"
- ▶ BBPATH = "\${TOPDIR}"
- ▶ BBFILES ?= ""
- ▶ BSPDIR := "/home/vagrant/rpi/"
- ▶ BBLAYERS ?= "\
- ▶ \${BSPDIR}/sources/poky/meta \
- ▶ \${BSPDIR}/sources/poky/meta-poky \
- ▶ \${BSPDIR}/sources/poky/meta-yocto-bsp \
- ▶ \${BSPDIR}/sources/meta-openembedded/meta-oe \
- ▶ \${BSPDIR}/sources/meta-openembedded/meta-multimedia \
- ▶ \${BSPDIR}/sources/meta-raspberrypi \



BUDUJEMY FIRMWARE

- ▶ bitbake rpi-basic-image
- ▶ bitbake rpi-hwup-image
- ▶ bitbake rpi-test-image

FW NA KARTE

- ▶ `sudo dd if=~/.rpi/rpi-build/tmp/deploy/images/raspberrypi3/rpi-basic-image-raspberrypi3.rpi-sdimg`
- ▶ `of=/dev/sdX bs=4M`

WŁĄCZENIE KAMERY

- ▶ Plik `conf/local.conf`
 - ▶ `GPU_MEM = "128"`
 - ▶ `VIDEO_CAMERA = "1"`
 - ▶ `CORE_IMAGE_EXTRA_INSTALL += "userland"`

DODANE OPENALPR

- ▶ **ściągnięcie dodatkowej warstwy**
 - ▶ `$ git clone https://github.com/maxinbjohn/meta-homeassistant-backup.git ../meta-homeassistant-backup`
- ▶ **modyfikacja `conf/bblayers.conf`**
 - ▶ `$ bitbake-layers add-layer ../meta-homeassistant-backup`
- ▶ **modyfikacja `conf/local.conf`**
 - ▶ `CORE_IMAGE_EXTRA_INSTALL += "openalpr"`

UTWORZENIE NOWEJ WARSTWY

- ▶ **utworzenie nowej warstwy**

- ▶ \$ bitbake-layers create-layer ../meta-main

- ▶ **dodanie nowej warstwy do conf/bblayers.conf**

- ▶ \$ bitbake-layers add-layer ../meta-main

- ▶ **utworzenie recepty main**

- ▶ \$ rm -rf ../meta-main/recipes-example/example

- ▶ \$ mkdir -p ../meta-main/recipes-example/main

- ▶ \$ touch ../meta-main/recipes-example/main/main_1.0.bb

PROBLEMY Z BUDOWANIEM

- ▶ Ubuntu 14 nie wyżej
- ▶ **wyczyszczenie pakietu** (main to nazwa czyszczonego pakietu)
 - ▶ \$ bitbake -c cleanall main
- ▶ uruchomienie powłoki deweloperskiej
 - ▶ \$ bitbake -c devshell main
- ▶ wyjście z powłoki
 - ▶ # exit
- ▶ wykonanie zadania kompilacji oraz zadań od niego zależnych
 - ▶ \$ bitbake -C compile main

PRZYDATNE INFO

- ▶ `do_build` - domyślne zadanie, wywołuje wszystkie normalne zadania
- ▶ `do_fetch` - pobiera źródła z `SRC_URI` do katalogu wskazywanego przez `DL_DIR`
- ▶ `do_unpack` - rozpakowuje źródła do katalogu wskazywanego przez `WORKDIR`
- ▶ `do_patch` - nakłada patche
- ▶ `do_configure` - konfiguruje źródła
- ▶ `do_compile` - kompiluje kod źródłowy, działa w katalogu roboczym `${B}`
- ▶ `do_install` - kopiuje pliki które mają trafić do pakietów do `${D}`
- ▶ `do_package` - rozdziela pliki dostarczone `${D}` na wiele pakietów

KONIEC

Janek Jabłonka