Установка дополнительных компонентов

1.
```
$ sudo apt-get update; sudo apt-get upgrade; sudo apt-get install libssl-dev python-is-python3 libgmp3-dev libmpfr-dev
```

2. Создаем папку для кросскомпилятора и устанавливаем.

```
$ mkdir gcc; cd ~/gcc
$ wget https://releases.linaro.org/components/toolchain/binaries/latest-7/arm-linux-gnueabihf/gcc-linaro-7.5.0-2019.12-x86_64_arm-linux-gnueabihf.tar.xz
$ tar –xvf gcc-linaro-7.5.0-2019.12-x86_64_arm-linux-gnueabihf.tar.xz
```

3. Устанавливаем SoCEDS.

```
$ sudo chmod +x SoCEDSSetup-20.1.0.711-linux.run
$ ./SoCEDSSetup-20.1.0.711-linux.run
```

После установки будет создана папка /home/Имя пользователя/intelFPGA/20.1

4. Заходим в /home/ИмяПользователя/intelFPGA/20.1/embedded и запускаем embedded_command_shell.sh. Это установит необходимые переменные окружения.

```
$ cd ~/my/intelFPGA/20.1/embedded
$ ./embedded_command_shell.sh
```

При дальнейшей работе не забываем запускать embedded_command_shell.sh для загрузки переменных окружения.

5. Заходим в /home/ИмяПользователя/intelFPGA/20.1/embedded/host_tools/linaro и запускаем install_linaro.sh:

```
$ ./install_linaro.sh
```

6. При компиляции не забываем устанавливать (**ВНИМАНИЕ**: локаль должна быть английской, иначе u-boot не собирается):

```
$ export LANG=en_US.UTF-8
$ export ARCH=arm
$ export CROSS_COMPILE=/путь к тулчейну/bin/arm-linux-gnueabihf- (префикс и путь должны указывать на нужный toolchain)
$ cd ~/my/intelFPGA/20.1/embedded
$ ./embedded_command_shell.sh
```

U-boot

1) Установка u-boot

```
$ git clone https://github.com/altera-opensource/u-boot-socfpga
$ cd u-boot-socfpga
$ cd ~/my/u-boot/u-boot-socfpga
git branch >>> * socfpga_v2024.07
```

2)

```
$ export ARCH=arm; export CROSS_COMPILE=~/my/all/gcc/gcc-linaro-7.5.0-2019.12-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-;

$ make clean & make mrproper
```

3) Генерация файлов BSP handof:

Берем проект квартус: /my/PC_50_M1/hps_isw_handoff/MCO_HPS_hps_0

```
$ cd ~/my/u-boot/u-boot-socfpga/board
$ mkdir agat/mco
$ cp -r terasic/de10-standard/* ~/my/u-boot/u-boot-socfpga/board/agat/mco/
```

В папке board/agat/mco/ будут находится папка qts, куда будут генерироваться handoff файлы, и файл MAINTAINERS.

```
$ nano ~/my/u-boot/u-boot-socfpga/board/agat/mco/MAINTAINERS
```

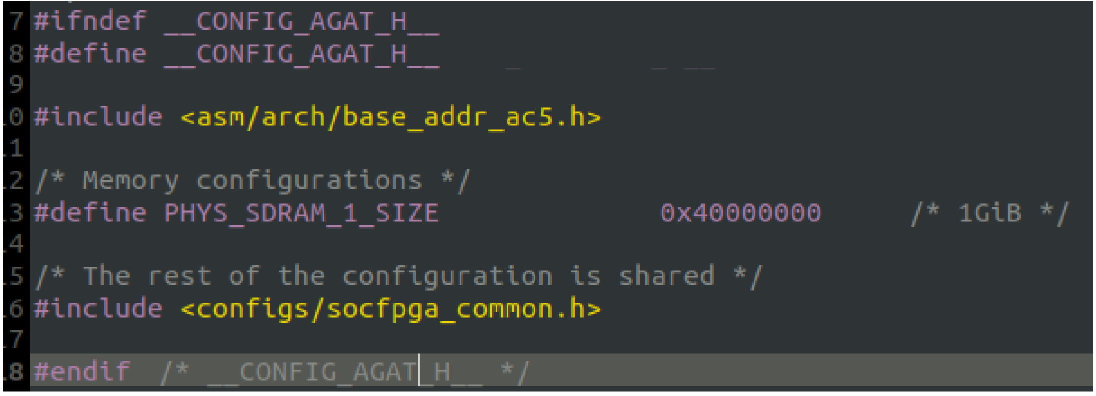
```
AGAT BOARD
M:     Humberto Naves <hsnaves@gmail.com>
S:     Maintained
F:     board/agat/mco/
F:     include/configs/agat.h
F:     configs/agat_defconfig
```

```
$ cd ~/my/u-boot/u-boot-socfpga/arch/arm/mach-socfpga/cv_bsp_generator

$ python cv_bsp_generator.py -i ~/my/PC_50_M1/hps_isw_handoff/MCO_HPS_hps_0 -o ../../../../board/agat/mco/qts
```

Результат в ~/my/u-boot/u-boot-socfpga/board/agat/mco/qts

4)

```
$ cd ~/my/u-boot/u-boot-socfpga/include/configs
$ cp -r socfpga_de10_standard.h agat.h
$ nano agat.h
```



```
#ifndef __CONFIG_AGAT_H__
#define __CONFIG_AGAT_H__

#include <asm/arch/base_addr_ac5.h>

/* Memory configurations */
#define PHYS_SDRAM_1_SIZE              0x40000000      /* 1GiB */

/* The rest of the configuration is shared */
#include <configs/socfpga_common.h>

#endif  /* __CONFIG_AGAT_H__ */
```

5)

```
$ cd ~/my/u-boot/u-boot-socfpga/configs
$ cp socfpga_de10_standard_defconfig socfpga_agat_mco_defconfig

$ nano socfpga_agat_mco_defconfig

   CONFIG_DEFAULT_DEVICE_TREE="socfpga_cyclone5_agat_mco"
   CONFIG_DEFAULT_FDT_FILE="socfpga_cyclone5_agat_mco.dtb"
   CONFIG_USB_GADGET_MANUFACTURER="agat"
```

```
6) $ nano ~/my/u-boot/u-boot-socfpga/arch/arm/mach-socfpga/Kconfig

   config TARGET_SOCFPGA_AGAT_MCO
         bool "AGAT MCO (Cyclone V)"
         select TARGET_SOCFPGA_CYCLONE5

   config SYS_VENDOR
         default "agat" if TARGET_SOCFPGA_AGAT_MCO

   config SYS_BOARD
         default "mco" if TARGET_SOCFPGA_AGAT_MCO

   config SYS_CONFIG_NAME
      default "socfpga_agat_mco" if TARGET_SOCFPGA_AGAT_MCO
```

```
7) $ cd ~/my/u-boot/u-boot-socfpga/arch/arm/dts
   $ cp socfpga_cyclone5_de10_standard.dts socfpga_cyclone5_agat_mco.dts
   $ nano socfpga_cyclone5_agat_mco.dts
         model = "Agat MCO";

   $ nano ~/my/u-boot-socfpga/arch/arm/dts/Makefile
```

в раздел *«dtb-$(CONFIG_ARCH_SOCFPGA)»* добавить сборку своего dts файла socfpga_cyclone5_agat_mco.dtb

```
8) $ make agat_defconfig
9) $ make menuconfig
   1. Boot options -> [*] Enable a default value for bootcmd
   (run fatscript) bootcmd value
   2. console -> loglevel(9)
   3. Networking support -> [*] Random ethaddr if unset
     Device Drivers -> GPIO Support -> Altera PIO driver
     Device Drivers -> LED Support -> Enable LED support
     Device Drivers -> LED Support -> LED support for GPIO-connected LEDs
     Device Drivers -> Network device support -> Enable RGMII

10) $ make savedefconfig
11) $ make
12) Также необходимо создать скрипт для u-boot

     $ nano u-boot.txt
          setenv ethaddr ca:ff:ee:12:34:56
          env set ipaddr 192.168.3.253;
          bridge enable;
          setenv loadbootsc no;
          env set fpgadata 0x10000000;
          fatload mmc 0:1 ${fpgadata} PC_50_M1.rbf;
          fpga load 0 ${fpgadata} ${filesize};
          setenv loadbootsc no;
          saveenv;
          run distro_bootcmd;
```

13) **$** ~/my/u-boot/u-boot-socfpga/tools/mkimage -A arm -O linux -T script -C none -a 0 -e 0 -n "Cyclone V script" -d u-boot.txt u-boot.scr

14) В дальнейшем нужны будут файлы **u-boot.scr** и **u-boot-with-spl.sfp**

Dts файл

```
$ dtc -I dts -O dtb -o agat_mco.dtb agat_mco.dts
```

Чтобы настроить триггер светодиода, необходимо отредактировать дерево устройств:

```
soc_leds: leds {
    compatible = "gpio-leds";

    led_hps0: hps0 {
          label = "hps_led0";
          linux,default-trigger = "heartbeat";
          gpios = <&hps_0_gpio1_porta 19 0>;
    }; //end hps0 (led_hps0)

    led_hps1: hps1 {
          label = "hps_led1";
          linux,default-trigger = "heartbeat";
          gpios = <&hps_0_gpio1_porta 5 1>;
    }; //end hps1 (led_hps1)
}; //end leds (soc_leds)
```

Из-за 0 и 1 будет моргать по-разному.

Linux

1) $ git clone http://github.com/altera-opensource/linux-socfpga

   ps git branch >> * socfpga-6.6.51-lts

2) $ export ARCH=arm; export CROSS_COMPILE=~/my/all/gcc/gcc-linaro-7.5.0-2019.12-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-
3) $ make clean
4) $ make mrpropper

5) $ cd linux-socfpga/
6) $ make socfpga_defconfig или $ make agat_defconfig (arch/arm/configs)
7) $ make menuconfig

1. General setup

-> General setup -> [ ] Automatically append version information to the version string (убираем звездочку, пока ядро находится в разработке. Когда мы будем уверены, что всё как надо, тогда возвращаем звездочку обратно.)

```
-> General setup -> (AGAT) Default hostname
-> General setup -> [*] Enable process_vm_readv/writev syscalls
-> General setup -> [*] uselib syscall (for libc5 and earlier)
-> General setup -> [*] CPU isolation
-> General setup -> <*> Kernel .config support
-> General setup -> [*] Enable access to .config through /proc/config.gz
-> General setup -> [*] Printk indexing debugfs interface
-> General setup -> [*] Control Group support (Memory controller, IO controller)
```

2. System Type

```
[*] MMU-based Paged Memory Management Support
[*] Require kernel to be portable to multiple machines
<*> FPGA DMA FIFO driver – не работает драйвер
```

3. Kernel Features

```
Kernel Features -> [*] Symmetric Multi-Processing
```

4. [*] Enable loadable module support

5. [*] Enable the block layer

6. [*] Networking support

```
-> Networking support -> Networking options
   <*> Packet socket
   <M>   Packet: sockets monitoring interface
   [*] Unix domain sockets
   <M>   UNIX: socket monitoring interface
   -*- Transformation migrate database
   <*> PF_KEY sockets
   [*]   PF_KEY MIGRATE
   [*] TCP/IP networking
   [*]   IP: multicasting
   [*]   IP: kernel level autoconfiguration
   [*]     IP: DHCP support
   [*]     IP: BOOTP support
   [*]     IP: RARP support
   <*>   IP: tunneling
   <*>   INET: socket monitoring interface
   <*>     UDP: socket monitoring interface
```

-> Networking support -> Networking options -> Network packet filtering framework (Netfilter) -> Core Netfilter Configuration -> Netfilter ingress support

-> Networking support -> Networking options-> Network packet filtering framework (Netfilter) -> Core Netfilter Configuration -> Netfilter Xtables support (required for ip_tables) -> <*>   "LED" target support, <*>   LOG target support

-> Networking support -> Networking options -> Network packet filtering framework (Netfilter) -> IP: Netfilter Configuration ->

```
<*> IPv4 socket lookup support
<M> IPv4 tproxy support
<*> IPv4 packet rejection
<*> ARP tables support
```

7. Device Drivers

-> Device Drivers -> Misc devices -> Altera FPGA firmware download module
-> Device Drivers -> Misc devices -> EEPROM support ->

```
          <*> I2C EEPROMs / RAMs / ROMs from most vendors
          <*> SPI EEPROMs (FRAMs) from most vendors
```

-> Device Drivers -> Network device support -> Network console logging support
-> Device Drivers -> Input device support -> Miscellaneous devices
-> Device Drivers -> I2C support -> I2C bus multiplexing support -> Multiplexer I2C Chip support

```
          <*> GPIO-based I2C arbitration
          <*> GPIO-based I2C multiplexer
```

-> Device Drivers -> SPI support -> Altera SPI Controller platform driver
-> Device Drivers -> SPI support -> GPIO-based bitbanging SPI Master

> Device Drivers > Watchdog Timer Support > Read different watchdog information through sysfs

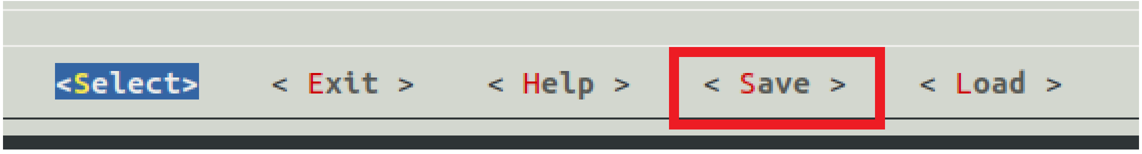
-> Device Drivers -> LED Support -> <*> LED Class Support (чтобы работал диод в режиме сердцебиение)
-> Device Drivers -> LED Support -> <*> LED Class Support -> <*> LED Flash Class Support
-> Device Drivers -> LED Support -> <*> LED Class Support -> [*] LED Class brightness_hw_changed attribute support

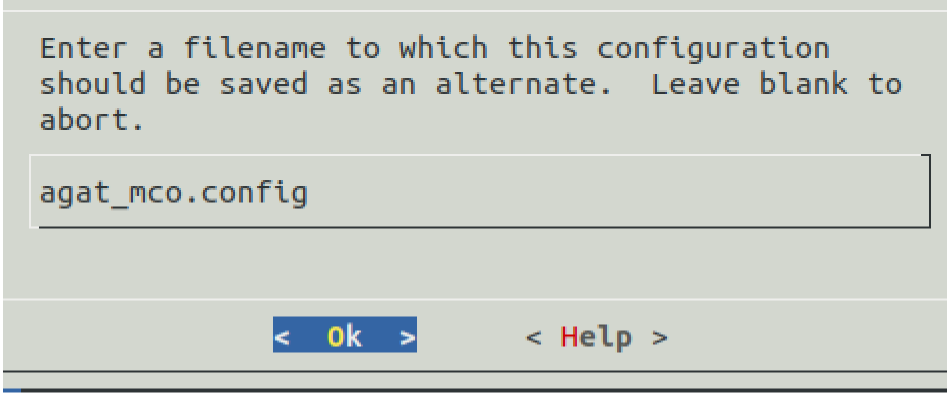
```
-> Device Drivers -> LED Support -> <*> LED Support for GPIO connected LEDs
-> Device Drivers -> LED Support -> <*> Userspace LED support
-> Device Drivers -> LED Support -> <*> LED Trigger support ->
           <*> LED Timer Trigger
           <*> LED One-shot Trigger
           <*> LED Heartbeat Trigger
           [*] LED CPU Trigger
           <*> LED activity Trigger
           <*> LED Default ON Trigger

-> Device Drivers -> DMA Engine support
           [*] DMA Engine debugging
           [*] DMA Engine verbose debugging
           <*> Altera / Intel mSGDMA Engine
           <*> DMA API Driver for PL330
           <*> DMA Test client

-> Device Drivers -> Industrial I/O support
     [*] Enable buffer support within IIO
     <*> IIO callback buffer used for push in-kernel interfaces
     <*>Industrial I/O buffering based on fifo
     -*- Enable IIO configuration via configfs
     [*] Enable triggered sampling support
     <*> Enable software IIO device support

-> Device Drivers -> <*> FPGA Configuration Framework ->
     [*]   FPGA Manager DebugFS
     <*> Altera Partial Reconfiguration IP -> <*> Platform support of Altera
     Partial Reconfiguration IP Core
```

8. Kernel hacking

```
     -> Kernel hacking -> printk and dmesg options
           [*] Show timing information on printks
           (7) Default console loglevel (1-15)
           (4) quiet console loglevel (1-15)
           (4) Default message log level (1-7)
           [*] Enable dynamic printk() support
           -*- Enable core function of dynamic debug support
           [*] Support symbolic error names in printf
           [*] Verbose BUG() reporting (adds 70K)
```

8)




Enter a filename to which this configuration should be saved as an alternate. Leave blank to abort.

agat_mco.config

```
9)   cp agat_mco.config arch/arm/configs/agat_mco_defconfig
10)   make agat_mco_defconfig
11)   $ make ARCH=arm LOCALVERSION= zImage -j 8
12)   $ export INSTALL_MOD_PATH=modules_install
13)   $ make modules
14)   $ make modules_install
15)   $ cd modules_install/lib/modules/6.6.51+; rm build && rm source; cd
   ../../../..
```

16)   создаем папку для сборки

```
cd ~/testboard/; mkdir 16; cd 16; mkdir sdfs
```

```
cp ~/путь/linux-socfpga/arch/arm/boot/zImage sdfs
```

ядро Kernel находится: ~/testboard/linux-socfpga/arch/arm/boot/zImage

модули находятся в: ~/testboard/linux-socfpga/modules_install/lib/modules/4.20.0+

дтб: ~/my/linux-socfpga/arch/arm/boot/dts/intel/socfpga

```
cd /sys/class/leds/hps_led1/
cat brightness
# echo "1">>brightness  // выключает диод
# echo "0">>brightness  // включает диод
```

BuildRoot

1) `$ git clone https://github.com/buildroot/buildroot`
   `git branch >> master`

2) `$  export  ARCH=arm;  export  CROSS_COMPILE=~/my/all/gcc/gcc-linaro-7.5.0-2019.12-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-`

3) `$ make clean`

4) `$ make socrates_cyclone5_defconfig`

5) `$ make menuconfig`

1. Target options

```
Target Architecture (ARM (little endian))
Target Architecture Variant (cortex-A9)
[*] Enable NEON SIMD extension support
[*] Enable VFP extension support
Target ABI (EABIhf)
Floating point strategy (NEON)
ARM instruction set (ARM)
Target Binary Format (ELF)
```

2. Toolchain

```
Toolchain type (Buildroot toolchain)
     *** Toolchain Buildroot Options ***
(buildroot) custom toolchain vendor name
c library (glibc)
     *** Kernel Header Options ***
Kernel Headers (Same as kernel being built) ---
Custom kernel headers series (6.6.x)
     *** Glibc Options ***
[ ] Enable compatibility shims to run on older kernels
[ ] Install glibc utilities
*** Binutils Options ***
Binutils Version (binutils 2.43.1) -
[ ] gprofng support
() Additional binutils options
     *** GCC Options ***
GCC compiler Version (gcc 13.x)
() Additional gcc options
[*] Enable C++ support
[ ] Enable Fortran support
[ ] Enable compiler OpenMP support
[ ] Enable graphite support
     *** Host GDB Options ***
[ ] Build cross gdb for the host
     *** Toolchain Generic Options ***
[ ] Copy gconv libraries
() Extra toolchain libraries to be copied to target
Target Optimizations
```

```
    Target linker options
        *** Bare metal toolchain ***
    [ ] Build bare metal toolchains
```

3. Build options

```
    Commands --->
        ((curly -q --ftp-pasv --retry 3 --connect-timeout 10) Curl command
        (wget --passive-ftp -nd -t 5 --no-check-certificate --connect-
        timeout=10) Wget command
        (svn --non-interactive --config-option servers:global:http-
        timeout=10) Subversion (svn) command
        (bzr) Bazaar (bzr) command
        (git) Git command
        (cvs) cVS command
        (cp) Local files retrieval command
        (scp -o ConnectTimeout=10) Secure copy (scp) command
        (sftp -o ConnectTimeout=10) Secure file transfer (sftp) command
        (hg) Mercurial (hg) command
        (gzip -d -c) zcat command
        (bzcat) bzcat command
        ((zcat) zcat command
        ((lzip -d -c) zcat command
        (zstdcat) zstdcat command
        () Tar options
    (/(/home/d/my/buildroot/configs/socrates_cyclone 5_defconfig) Location to
    save buildroot config
    ($(($(TOPDIR)/dl) Download dir
    ($(BASE_DIR)/host) Host dir
    Mirrors and Download locations --->
        ( http://ftp.gnu.org/gnu ) Primary download site
        [] Only allow downloads from primary download site (NEW)
        ( https://sources.buildroot.net ) Backup download
        ( https://cdn.kernel.org/pub ) Kernel.org mirror
        ( https://ftp.gnu.org ) GNU Software mirror
        ( http://rocks.moonscript.org ) LuaRocks mirror
        ( https://cpan.metacpan.org ) CPAN mirror (Perl packages)

    (0) Number of jobs to run simultaneously (0 for auto)
    [ ] Enable compiler cache
    [ ] build packages with debugging symbols
    [ ] build packages with runtime debugging info
    [*] strip target binaries
    ()
    executables that should not be stripped
    () directories that should be skipped when stripping
    gcc optimization level (optimization level 2)
    [ ] build packages with link-time optimisation
    [ ] Enable google-breakpad support (NEW)
    libraries (shared only)
```

```
($(CONFIG_DIR)/local.mk) location of a package override file
() global patch and hash directories
Advanced - - ->
[ ] Build Y2038-ready code
*** Security Hardening Options ***
-*- Build code with PIC/PIE
Stack Smashing Protection (-fstack-protector-strong) ---
RELRO Protection (Full)
Buffer-overflow Detection (FORTIFY_SOURCE) (Conservative)
```

4. System configuration

```
Root FS skeleton (default target skeleton)
(MCO) System hostname
(Welcome to AGAT!) System banner
Passwords encoding (sha-256)
Init system (BusyBox)
/dev management (Dynamic using devtmpfs only)
(system/device_table.txt) Path to the permission tables
[] support extended attributes in device tables
[*] Use symlinks to /usr for /bin, /sbin and /lib
[*] Enable root login with password
(1234) Root password
/bin/sh (busybox' default shell)
[*] Run a getty (login prompt) after boot
remount root filesystem read-write during boot
(eth0) Network interface to configure through DHCP
(/bin:/sbin:/usr/bin:/usr/sbin) Set the system's default PATH
[*] Purge unwanted locales
(C en_US) Locales to keep
() Generate locale data
[ ] Enable Native Language Support (NLS)
[ ] Install timezone info
() Path to the users tables
(/home/d/my/sdim/10./sdfs/LinuxConf) Root filesystem overlay directories
() Custom scripts to run before commencing the build
() Custom scripts to run before creating filesystem images
Custom scripts to run inside the fakeroot environment
(support/scripts/genimage.sh) Custom scripts to run after
creating filesystem images
() Extra arguments passed
to custom scripts
() Extra arguments passed to POST_IMAGE_SCRIPT
```

5. Kernel -> No

6. Target packages

```
-*-BusyBox
(package/busybox/busybox.config) BusyBox configuration file to use?
( ) Additional BusyBox configuration fragment files
Show packages that are also provided by busybox
```

Individual binaries
[ ] Install the watchdog daemon startup script

Compressors and decompressors ---> bzip2
Debugging, profiling and benchmark--->

gdb (gdbserver, full debugger, TUI support), spidev_test

Development tools--->

git, libtool, make, subversion, tree

Filesystem and flash utilities--->

cifs-utils, f2fs-tools, mmc-utils, mtd, jffs2 and ubi/ubifs tools(flashcp, flash_erase, flash_lock, flash_otp_info, flash_otp_write, flash_unlock, jffs2dump, mkfs.jffs2, mkfs.ubifs, mtd_debug, nanddump, nandtest, handwritten, mtd info, ubiattach, ubicrc32, ubidetach, ubiformat, ubihealth, ubimkvol, subinfo, ubinize, ubirename, ubirmvol, ubirsvol, ubiupdatevol, ubi block), mtools

Hardware handling--->

altera-stapl, flashrom, libubootenv, libuio, linux-serial-test, linuxconsoletools, lsuio, memtester, openfpgaloader, openocd (Altera USB-Blaster II Compatible, Altera USB-Blaster), rs485conf, setserial, spi-tools, statserial, sysstat

Libraries--->

Compression and decompression --->

Libarchive, lz4, lzo , zlib support

Crypto--->

CA Certificates, libassuan, libgcrypt, libgpg-error, libgpgme, libksba, libssh (openssl), libssh2 (openssl), ustream-ssl

Filesystem --->

libsysfs

Hardware handling --->

c-periphery, dtc (libfdt) ,dtc programs, hackrf, libaio, libftdi (C++ bindings), libgpiod (install tools), libiio (и всё, что к нему относится), Libserial, libserialport, libsoc, libusb, libusb-compat

Networking---> c-ares

Miscellaneous--->

haveged (является базовым демоном для генерации случайных чисел)

Networking applications--->

Autossh, can-utils, dhcpd (демон для работы с сетью и получения сетевых реквизитов по протоколу DHCP), dropbear (демон для организации SSH-сервера), iputils ifupdown scripts (для работы с сетевыми интерфейсами), iperf, iperf3 (для проведения бенчмарка пропускной способности по сети), sshpass

Package managers---> opkg (gnupg support)
Shell and utilities---> gnupg (AES support), sudo
System tools--->

attr, cpuload, deamon, efibootmgr (efivar), htop, keyutils, kmod, util-linux(libblkid, libmount, libsmartcols, libuuid, fsck, kill, line, logger, login, lslogins, lsmem, mesg, more, mount/unmount, mountpoint, su, switch_root, wdctl), Watchdog

Text editors and viewers---> mc

7. Filesystem images

```
[ ] axfs root filesystem
[ ] btrfs root filesystem
[ ] cloop root filesystem for the target device
[ ] cpio the root filesystem (for use as an initial RAM filesystem)
[ ] cramfs root filesystem
[ ] erofs root filesystem
[*] ext2/3/4 root filesystem
ext2/3/4 variant (ext4)  --->
(rootfs) filesystem label
(200M) exact size
(0)   exact number of inodes (leave at 0 for auto calculation)
(256) inode size
(5)   reserved blocks percentage
(-O ^64bit) additional mke2fs options
Compression method (no compression)  --->
[ ] f2fs root filesystem
*** initramfs needs a Linux kernel to be built ***
[ ] jffs2 root filesystem
[ ] oci image
[ ] squashfs root filesystem
[*] tar the root filesystem
Compression method (no compression)  --->
()    other random options to pass to tar
[ ] ubi image containing an ubifs root filesystem
[ ] ubifs root filesystem
[ ] yaffs2 root filesystem
```

8. Bootloaders (убираем везде звездочки)

9. Host utilities ---> host e2tools

p.s.1 Наложение корневой файловой системы

https://stackoverflow.com/questions/37046170/static-network-interface

System Configuration -> Root Filesystem Overlay ->
…пишем..путь..до../sdfs/LinuxConf

/home/d/my/sdim/10./sdfs/LinuxConf

- **LinuxConf -> etc-> network -> interfaces**

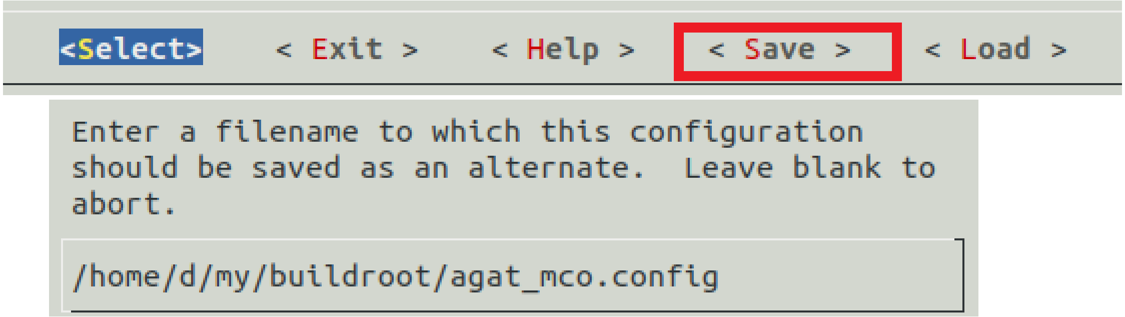
```
# interface file auto-generated by buildroot
auto lo
iface lo inet loopback

allow-hotplug eth0
iface eth0 inet static
address 192.168.3.239
netmask 255.255.255.0
gateway 192.168.3.240
auto eth0
```

- **LinuxConf -> etc-> ssh ->**

```
#	$OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/bin:/sbin:/usr/bin:/usr/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
```

```
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication.  Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
#UsePAM no

#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
#X11Forwarding no
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
#PrintMotd yes
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# override default of no subsystems
Subsystem    sftp   /usr/libexec/sftp-server
```

```
# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
```

6) 

Enter a filename to which this configuration should be saved as an alternate. Leave blank to abort.

/home/d/my/buildroot/agat_mco.config

7) $ make busybox-source

8) $ make busybox-menuconfig
Login/Password Management Utilities -> Use internal password and group functions rather than system functions

9) $ make all -j 8

результат сборки: /buildroot/output/images/rootfs.tar

идем в каталог ~/my/buildroot/output/images

копируем в ~/my/sdim/1. 26042025-1/rootfs архив

tar -xvf rootfs.tar

rm rootfs.tar

cd rootfs/bin

sudo chown root * -R - тогда можно войти в систему без ошибок

если после распаковки архива зайти в папку rootfs/bin и написать **sudo chown root * -R** , тогда можно войти в систему без ошибок

заходим в /home/darya/testboard/12/buildroot/output/images

распаковываем архив в ~/testboard/ 16

создастся папка rootfs

sudo ./makeSDimage