wget http://ftp.gnu.org/pub/gnu/gettext/gettext-0.19.tar.gz ——— MAKED. — **INSTALLED** **sudo apt-get install gettext**

tar xvf gettext-0.19.tar.gz

cd gettext-0.19

First, suppress two invocations of test-lock which on some machines can loop forever:

```
sed -i '/^TESTS =/d' gettext-runtime/tests/Makefile.in &&
sed -i 's/test-lock..EXEEXT.//' gettext-tools/gnulib-
tests/Makefile.in
```

Prepare Gettext for compilation:

```
./configure --prefix=/usr    \
            --disable-static \
            --docdir=/usr/share/doc/gettext-0.19
```

Compile the package:

```
make
```

To test the results (this takes a long time, around 3 SBUs), issue:

```
make check
```

Install the package:

```
make install
chmod -v 0755 /usr/lib/preloadable_libintl.so
```

—> altra soluzione sudo apt-get install gettext

ftp://sourceware.org/pub/libffi/libffi-3.2.1.tar.gz — MAKED —

INSTALLED

tar xvf libffi-3.2.1.tar.gz

cd libffi-3.2.1

```
sed -e '/^includesdir/ s/$(libdir).*$/$(includedir)/' \
    -i include/Makefile.in &&

sed -e '/^includedir/ s/=.*$/=@includedir@/' \
    -e 's/^Cflags: -I${includedir}/Cflags:/' \
    -i libffi.pc.in        &&

./configure --prefix=/usr --disable-static &&
make

sudo make install
```

http://www.cmake.org/files/v3.8/cmake-3.8.2.tar.gz

—––– making. – **INSTALLED WITH**

sudo apt-get install cmake

tar xvf cmake-3.8.2.tar.gz

cd cmake-3.8.2

```
sed -i '/CMAKE_USE_LIBUV 1/s/1/0/' CMakeLists.txt     &&
sed -i '/"lib64"/s/64//' Modules/GNUInstallDirs.cmake &&

./bootstrap --prefix=/usr        \
            --system-libs        \
            --mandir=/share/man  \
            --no-system-jsoncpp  \
            --no-system-librhash \
            --docdir=/share/doc/cmake-3.8.2 &&
make

make install
```

——> ALTRA SOLUZIONE : sudo apt-get install cmake

http://www.mega-nerd.com/libsndfile/files/libsndfile-1.0.28.tar.gz — MAKED

tar xvf libsndfile-1.0.28.tar.gz

cd libsndfile-1.0.28

```
./configure --prefix=/usr    \
            --disable-static \
            --docdir=/usr/share/doc/libsndfile-1.0.28 &&
make

make check

make install
```

http://www.kernel.org/pub/linux/bluetooth/sbc-1.3.tar.xz. —

MAKED — INSTALLED

tar xvf sbc-1.3.tar.xz

cd sbc-1.3

```
./configure --prefix=/usr --disable-static --disable-
tester &&
make

make install
```

https://github.com/libical/libical/releases/download/v2.0.0/libical-2.0.0.tar.gz ——————— MAKED — INSTALLING
tar xvf libical-2.0.0.tar.gz

cd libical-2.0.0

```
mkdir build &&
cd build &&

cmake -DCMAKE_INSTALL_PREFIX=/usr       \
      -DCMAKE_BUILD_TYPE=Release        \
      -DSHARED_ONLY=yes                 \
      .. &&
make

sudo make install
```

http://ftp.gnome.org/pub/gnome/sources/glib/2.52/glib-2.52.2.tar.xz ——————— MAKING ERROR - INSTALLED WITH sudo apt-get install glib-2.0

tar xvf glib-2.52.2.tar.xz

cd glib-2.52.2

```
./configure --prefix=/usr --with-pcre=system &&
make -f make


sudo make install
```

—> altra soluzione : sudo apt-get install libglib2.0-dev

http://dbus.freedesktop.org/releases/dbus/dbus-1.10.18.tar.gz — MAKING

tar xvf dbus-1.10.18.tar.gz

cd dbus-1.10.18

```
groupadd -g 18 messagebus &&
useradd -c "D-Bus Message Daemon User" -d /var/run/dbus \
        -u 18 -g messagebus -s /bin/false messagebus

./configure --prefix=/usr                        \
            --sysconfdir=/etc                    \
            --localstatedir=/var                 \
            --disable-doxygen-docs               \
            --disable-xml-docs                   \
            --disable-static                     \
            --docdir=/usr/share/doc/dbus-1.10.18 \
            --with-console-auth-dir=/run/console \
            --with-system-pid-file=/run/dbus/pid \
            --with-system-socket=/run/dbus/
system_bus_socket


make

sudo make install

dbus-uuidgen --ensure
```

http://dbus.freedesktop.org/releases/dbus-glib/dbus-glib-0.108.tar.gz. ——— making ERROR WAITING FOR DBUS-1

tar xvf dbus-glib-0.108.tar.gz

cd dbus-glib-0.108

```
./configure --prefix=/usr     \
            --sysconfdir=/etc \
            --disable-static


make

make install
```

wget http://www.kernel.org/pub/linux/bluetooth/bluez-5.45.tar.xz ———— MAKING error `GLib >= 2.28 is required`

tar xvf bluez-5.45.tar.xz

cd bluez-5.45

```
./configure --prefix=/usr          \
            --sysconfdir=/etc      \
            --localstatedir=/var   \
            --enable-library      \
            --disable-systemd     &&
make
```

Now, as the *root* user:

```
make install &&
ln -svf ../libexec/bluetooth/bluetoothd /usr/sbin
```

Install the main configuration file as the *root* user:

```
install -v -dm755 /etc/bluetooth &&
install -v -m644 src/main.conf /etc/bluetooth/main.conf
```

If desired, install the API documentation as the *root* user:

```
install -v -dm755 /usr/share/doc/bluez-5.45 &&
install -v -m644 doc/*.txt /usr/share/doc/bluez-5.45
```

Configuring BlueZ

Configuration Files

`/etc/bluetooth/main.conf` is installed automatically during the install. Additionally, there are three supplementary configuration files. /etc/sysconfig/bluetooth is installed as a part of the boot script below. In addition, you optionally can install the following, as the *root* user:

```
cat > /etc/bluetooth/rfcomm.conf << "EOF"
# Start rfcomm.conf
# Set up the RFCOMM configuration of the Bluetooth
subsystem in the Linux kernel.
# Use one line per command
# See the rfcomm man page for options


# End of rfcomm.conf
EOF
cat > /etc/bluetooth/uart.conf << "EOF"
# Start uart.conf
# Attach serial devices via UART HCI to BlueZ stack
# Use one line per device
# See the hciattach man page for options
# End of uart.conf
EOF
```

Boot Script

To automatically start the **bluetoothd** daemon when the system is rebooted, install the /etc/rc.d/init.d/bluetooth bootscript from the blfs-bootscripts-20170611 package.

```
make install-bluetooth
```

http://anduin.linuxfromscratch.org/BLFS/blfs-bootscripts/blfs-bootscripts-20170611.tar.xz

tar xvf blfs-bootscripts-20170611.tar.xz

cd blfs-bootscripts-20170611

```
cp ~/bluetooth /etc/rc.d/init.d/bluetooth
```

To install your own script, copy it to /etc/init.d, and make it executable.

```
    sudo cp ~/blfs-bootscripts-20170611/blfs/init.d/bluetooth /etc/rc.d/init.d/bluetooth
    sudo chmod +x /etc/init.d/bluetooth
```

To make the script run with the start argument at the end of the start sequence, and run with the stop argument at the beginning of the shutdown sequence:

```
    sudo update-rc.d bluetooth defaults
```

```
wget http://www.mega-nerd.com/libsndfile/files/libsndfile-1.0.28.tar.gz

tar xvf libsndfile-1.0.28.tar.gz

cd libsndfile-1.0.28
```

```
./configure --prefix=/usr    \
            --disable-static \
            --docdir=/usr/share/doc/libsndfile-1.0.28 &&
make

make install
```

```
wget https://s3.amazonaws.com/json-c_releases/releases/json-c-0.12.1.tar.gz.         —making

tar xvf json-c-0.12.1.tar.gz

cd json-c-0.12.1
```

sudo apt-get install m4

```
sudo apt-get install autoconf
```

```
sed -i s/-Werror// Makefile.in tests/Makefile.in &&
./configure --prefix=/usr --disable-static       &&
make -j1

sudo make install
```

ownloading m4:

Run the command below,

```
wget ftp://ftp.gnu.org/gnu/m4/m4-1.4.10.tar.gz
```

Extracting files from the downloaded package:

```
tar -xvzf m4-1.4.10.tar.gz
```

Now, enter the directory where the package is extracted.

```
cd m4-1.4.10
```

Configuring m4:

```
./configure --prefix=/usr/local/m4
```

Replace "/usr/local/m4" above with the directory path where you want to copy the files and folders. Note: check for any error message.

Compiling m4:

```
make
```

Note: check for any error message.

Installing m4:

As root (for privileges on destination directory), run the following.

With sudo,

```
sudo make install
```

Without sudo,

```
make install
```

wget http://freedesktop.org/software/pulseaudio/releases/pulseaudio-10.0.tar.xz.

— MAKED — installing

tar xvf pulseaudio-10.0.tar.xz

cd pulseaudio-10.0

--disable-bluez5: This switch prevents a runtime error if dbus-1.10.18 and SBC-1.3 are both installed but BlueZ-5.45 is not installed. Remove this if you have installed all three packages.

```
            --disable-bluez5      \
./configure --prefix=/usr          \
            --sysconfdir=/etc      \
            --localstatedir=/var \
            --disable-bluez4       \
            --disable-rpath        &&
make


sudo make install
```

```
rm /etc/dbus-1/system.d/pulseaudio-system.conf
```

```
sed -i '/load-module module-console-kit/s/^/#/' /etc/
pulse/default.pa
```

You may also have to configure the audio system. You can start pulseaudio in command line mode using `pulseaudio -c` and then list various information and change settings. See `man pulse-cli-syntax.`