Bluetooth:

1- Configurations through buildroot menu

- Target packages >> Hardware handling >> Firmware >> rpi-bt-firmware
- Target packages >> Networking applications >> bluez-tools
- Target packages >> Networking applications >> bluez-utils 5.x (and all under it)
- Target Packages >> Audio and Video applications >> bluez-alsa

Note: This last one is to enable audio through bluetooth not to enable the bluetooth itself

2- Changes made to the post-build script

- N/A

3- Changes made to the pre-build script

- N/A

4- Script added to the overlay

This script is to be run by a daemon to search and connect to bluetooth devices.

- The script initializes the bluetooth module on the rpi one time only.
- Then the script powers on the bluetooth and starts searching for any bluetooth device after that the script read the log from the previous operation to get the MAC addresses of any nearby device and stores them in a file, if there aren't any devices the script restarts the search again.
- The script tries to pair with any of the devices in the file by trusting the device then trying to pair then accepting the pair then connecting.
- The script then checks if the previous operation succeeded -through the log- and if it did succeed, it plays a message "Bluetooth device connected", and if it didn't it tries the previous operation again with a different device.
- After connecting the script checks if the device is still connected by checking for the connected devices info, and if it is still connected it does this operation again and again till the device is not connected anymore which leads to the script to start the search from the beginning after playing "Bluetooth device disconnected" message.
- In the main mp3 player script if there is a bluetooth dvice connected we replace the line "mpg123 $Song" with
  "mpg123" -D bluealsa:DEV="**MAC**"PROFILE=a2dp,HCI=hci0 $Song"

5- Problems we faced

The main problem with the bluetooth script is the method the script takes to take any action towards the bluetooth. For example: we couldn't use the "bluetotothctl" comand directly because it would take control over the terminal and waits for an input from the user which isn't right because the script is required to do its job in the background without any interference from the user, also we couldn't use "bluetotothctl &" command to make it run in the background because it requires inputs which will make the process killed instantly.

The solution was to use "coproc bluetotothctl" which runs the script in the background and setting up pipes connected to both its stdin and stdout where we could give inputs "scan on-trust-pair ...." and read the output of each operation.

Audio output device:

1- Configurations through buildroot menu

- N/A

2- Changes made to the post-build script

- N/A

3- Changes made to the pre-build script

- N/A

4- Script added to the overlay

This script is to be run by a daemon to detect the connected audio device.

- The script starts by checking the output from the bluetooth script if it is true then it notfries the mp3 player that a bluetooth device is connected.
- The script then checks for an HDMI cable connection using "tvservice" command and if it is true the audio is directed to the HDMI device through "amixer" command and notify the mp3 player.
- If neither the bluetooth device nor the HDMI device are connected then it notifies the mp3 player that it is the audio jack that is connected.
- The script plays a message for every device connected.

5- Problems we faced

There were two problems with that script.

- For the HDMI the "tvservice" command is not available on buildroot so we had to clone the repo "userland" where the command was found and compile it manually to generate the binary to be used by the script.
- For the audio jack part, we couldn't figure out a way to detect if there is an audio jack connected or not.