1/ What is gstreamer?

- Is a framework for creating streaming media applications
- The fundamental design comes from the video pipeline at Oregon Graduate Institute, as well as some ideas from DirectShow.
- Possible to write any type of streaming multimedia application.
- Advantage: the pluggable components can be mixed and matched into arbitrary pipelines. This pipeline defines the flow of the data.

2/ Three main part of gstreamer:

2.1/ Applications and tools:

GStreamer core provided gst-inspect to query information of plugins; gst-launch to create pipeline from elements of plugins.

2.2/ Core:

Provide framework for plugins, data flow and media type handling/negotiation. Provide API for application.

2.3/ Plugins:

There are many types of plugins, basic plugins and plugins from 3rd party.

3/ Basic conceps of gstreamer**:**

3.1/ Elements

Element linked together and let data go through them

Chaining several element together, a pipeline for a specific task is created.

Input and output of element are called pads.

Sink pad is input of element, where receive data from other element

Source pad is output element, where send data to other element

One lelement have 4 state is: NULL/ READY/ PAUSE/ PLAYING

3.1.1/ Element and plugin

+ Use gst_element_set_state () to change state for element

+ Element class will be in-charge for change state.

+ In case setting an element from NULL to PLAYING, element will internally change to READY -> PAUSE before PLAYING

+ In case setting an element from PLAYING to NULL, element will internally change to PAUSE -> READY before NULL

Plugins: A plugin is essentially a loadable block of code (shared object file or a dynamically linked library). A single plugin may contain the implementation of several elements, or just a single one

3.2/ Pad

This is input and output of a element. The element type can be classified depend on the number of pad.

+ Source element: have only one source pad.

+ Filer and filter –like element: have two element is sink pad and source pad.

+ Sink element: only hae sink pad (often at the end oof pipeline).

3.2.1/ Capabilities of pad

Describe the type of data that is streamed between two pads, or that one pad (template) supports

3.2.2/ Caps negotiation

Capabilities of 2 pads are compatible each other.

3.3/ Bin

Bin elements: contain other elements.

There are 2 types: responsible for synchronization of the elements that they contain so that data flows smoothly automatically add other elements to the bin and links them together

3.4/ GstMiniObject, Buffers and Events

GstMiniObject: is the structure used to hold chunks of data (all streams of data in GStreamer) Buffers: contains a chunk of audio or video data that flows from one element to another.

Events: contain information on the state of the stream flowing between the two linked pads. Events are used to indicate: a media type, the end of a media stream, seeking and so on.

4/ Communication on gstreamer:

Buffers: are objects for passing streaming data between elements in the pipeline. Buffers always travel from sources to sinks

Events: are objects sent between elements or from the application to elements. Events can travel upstream and downstream.

Messages: are objects posted by elements on the pipeline's message bus, where they will be held for collection by the application. Messages are used to transmit information such as errors, tags, state changes, buffering state, redirects etc. from elements to the application in a thread-safe way.

Queries: allow applications to request information such as duration or current playback position from the pipeline. Queries are always answered synchronously. Elements can also use queries to request information from their peer elements

4/ Create pipeline

Pipeline can be created by using gst-launch tool to connect for elements on system. Elements will connect each other by “!”. Element before “!”, we call upstream element. Element after “!”, we call downstream element. Besides, we can assign type for negotiation by using capsfilter element in case element have many types.

In case you don’t have knowledge about the video file, can use playbin: # gst-launch-1.0 -v playbin uri=file:///path/to/somefile.mp4
Playbin will detect suitable elements in filesystem for playback automatically.

5/ Build gst in yocto

We can rebuilt packages of GStreamer by following command line:
#bitbake gstreamer1.0xxx –C configure

6/ Install built packages into filesystem

Can find them on *$build/tmp/work/aarch64-poky-linux/gstreamer1.0xxx/<version>/deploy-ipks/aarch64/*

To install for these plugins use “opkg install” plugin.ipk