NAME

ofw_graph, **ofw_graph_get_port_by_idx**, **ofw_graph_port_get_num_endpoints**, **ofw_graph_get_endpoint_by_idx**, **ofw_graph_get_remote_endpoint**, **ofw_graph_get_remote_parent**, **ofw_graph_get_device_by_port_ep** - Helpers for the graph bindings

SYNOPSIS

#include <dev/ofw/openfirm.h>
#include <dev/ofw/ofw_graph.h>

phandle_t
ofw_graph_get_port_by_idx(*phandle_t node*, *uint32_t idx*);

size_t
ofw_graph_port_get_num_endpoints(*phandle_t port*);

phandle_t
ofw_graph_get_endpoint_by_idx(*phandle_t port*, *uint32_t idx*);

phandle_t
ofw_graph_get_remote_endpoint(*phandle_t endpoint*);

phandle_t
ofw_graph_get_remote_parent(*phandle_t remote*);

device_t
ofw_graph_get_device_by_port_ep(*phandle_t node*, *uint32_t port_id*, *uin32_t ep_id*);

DESCRIPTION

The ofw_graph functions are helpers to parse the DTS graph bindings

ofw_graph_get_port_by_idx() return the port with id *idx*. It will first check node named *port@idx* and then fallback on checking the *ports* child for a child node matching the id. If no ports matching *idx* is found the function return 0.

ofw_graph_port_get_num_endpoints() returns the number of endpoints a port node have.

ofw_graph_get_endpoint_by_idx() return the endpoint with id *idx*. It will first check if there is a single child named *endpoint* and returns it if there is. If there is multiple endpoints it will check the *reg* property and returns the correct *phandle_t* or 0 if none match.

ofw_graph_get_remote_endpoint() returns the *remote-endpoint* property if it exists or 0.

ofw_graph_get_remote_parent() returns the device node corresponding to the *remote-endpoint* phandle or 0 if none. **ofw_graph_get_device_by_port_ep**() returns the device associated with the port and endpoint or *NULL* if none. The device driver should have called **OF_device_register_xref**() before.

HISTORY

The **ofw_graph** functions first appeared in FreeBSD 13.0. The **ofw_graph** functions and manual page were written by Emmanuel Vadot <*manu@FreeBSD.org*>.