

```

int dsp_shutdown()
{
    mcapi_status_t status;
    mcapi_request_t request;

    mcapi_pktchan_release(data_buffer, &status);

    mcapi_pktchan_release(code_buffer, &status);

    mcapi_pktchan_rcv_close_i(data_chan, &request, &status);
    CHECK_STATUS(status);

    mcapi_sclchan_rcv_close_i(command_chan, &request, &status);
    CHECK_STATUS(status);

    mcapi_pktchan_send_close_i(data_send_chan, &request, &status);
    CHECK_STATUS(status);

    mcapi_endpoint_delete(dsp_command_endpoint, &status);
    CHECK_STATUS(status);

    mcapi_endpoint_delete(dsp_data_endpoint, &status);
    CHECK_STATUS(status);

    mcapi_endpoint_delete(dsp_data_send_endpoint, &status);
    CHECK_STATUS(status);

    return status;
}

int receive_dsp_cmd()
{
    mcapi_uint32_t cmd;
    mcapi_status_t status;

    cmd = mcapi_sclchan_rcv_uint32(command_chan, &status);
    CHECK_STATUS(status);
}

int dsp_command_loop()
{
    int command = DSP_READY;
    mcapi_status_t status;

    dsp_initialize();

    while (command != DSP_TERMINATE) {
        switch (command) {
            case DSP_DATA:
                mcapi_pktchan_rcv(data_chan, &data_buffer, &data_size, &status);
                CHECK_STATUS(status);
                break;
            case DSP_CODE:
                mcapi_pktchan_rcv(data_chan, &code_buffer, &code_size, &status);
                CHECK_STATUS(status);
                /* Copy code to from buffer to local execute memory */
                break;
            case DSP_EXECUTE:
                /* Tell DSP to Execute - Assume code writes to result_buffer */

```