```
	mcapi_sclchan_send_uint32(command_chan, cmd, &status);
	CHECK_STATUS(status);
}

void read_data(void **dst, int *size)
{
	mcapi_status_t status;
	mcapi_pktchan_recv(data_recv_chan, dst, size, &status);
	CHECK_STATUS(status);
}

void shutdown_comms()
{
	mcapi_status_t status;
	mcapi_request_t request;

	mcapi_pktchan_recv_close_i(data_recv_chan, &request, &status);
	CHECK_STATUS(status);
	mcapi_pktchan_send_close_i(data_chan, &request, &status);
	CHECK_STATUS(status);
	mcapi_sclchan_send_close_i(command_chan, &request, &status);
	CHECK_STATUS(status);

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

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

	mcapi_endpoint_delete(data_recv_endpoint, &status)
	CHECK_STATUS(status);
}

void *allocate(int size) {
	/* Routine to allocate memory */
}

int perform_multimedia_function(void *code, int code_size, void *data, int
data_size)
{
	void* result;
	int size = 0;
	initialize_comms();
	send_dsp_cmd(DSP_DATA);
	send_data(data, data_size);
	send_dsp_cmd(DSP_CODE);
	send_data(code, code_size);
	send_dsp_cmd(DSP_EXECUTE);
	read_data(&result, &size);
	send_dsp_cmd(DSP_TERMINATE);
	shutdown_comms();
	return COMPLETED;
}

/*
 * MCAPI 2.000 Multimedia Use Case
 * dsp.c
 *
 */

/* DSP Code */
```