```
      DSP_READY,
      DSP_DATA,
      DSP_CODE,
      DSP_TERMINATE,
      DSP_EXECUTE,
      COMPLETED
};

#define DOMAIN_0        0

#define PORT_COMMAND    0
#define PORT_DATA       1
#define PORT_DATA_RECV  2

#endif /* SHARED_H_ */

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

/* General Purpose Processor Code */
#include "shared.h"
#include "mcapi.h"

/* Predefined Node numbering */
#define GPP_1 0  /* 0-63 reserved for homogenous MC */
#define DSP_1 64 /* 64+ used for DSP nodes */

/* Predefined Port numbering */
#define PORT_COMMAND 0
#define PORT_DATA 1
#define PORT_DATA_REC 2

mcapi_endpoint_t command_endpoint;
mcapi_endpoint_t data_endpoint;
mcapi_endpoint_t remote_command_endpoint;
mcapi_endpoint_t remote_data_endpoint;
mcapi_endpoint_t data_recv_endpoint;
mcapi_endpoint_t remote_data_recv_endpoint ;

mcapi_pktchan_send_hndl_t data_chan;
mcapi_sclchan_send_hndl_t command_chan;
mcapi_pktchan_recv_hndl_t data_recv_chan;

void *data;
int data_size;
void *code;
int code_size;

void initialize_comms()
{
      mcapi_node_t gp_node = 0;
      mcapi_param_t mcapi_parameters;
      mcapi_info_t mcapi_info;
      mcapi_status_t status;
      mcapi_request_t request;

      mcapi_initialize(DOMAIN_0, gp_node, &mcapi_parameters, &mcapi_info,
&status);
```