```
   mcapi_initialize(ENGINE_DOMAIN, CNTRL_NODE, &mcapi_parameters,
&mcapi_info, &err);
   CHECK_STATUS(err);

   // first create two local endpoints
   tpu_endpt = mcapi_endpoint_create(CNTRL_PORT_TPU, &err);
   CHECK_STATUS(err);

   sig_endpt = mcapi_endpoint_create(CNTRL_PORT_SIG, &err);
   CHECK_STATUS(err);

   // now we get two remote endpoints
   mcapi_endpoint_get_i(ENGINE_DOMAIN, TPU_NODE, TPU_PORT_CNTRL,
&tpu_remote_endpt, &r1, &err);
   CHECK_STATUS(err);

   mcapi_endpoint_get_i(ENGINE_DOMAIN, SIG_NODE, SIG_PORT_CNTRL,
&sig_remote_endpt, &r2, &err);
   CHECK_STATUS(err);

   // wait on the endpoints
   while(!((mcapi_test(&r1, &tSize, &err))) && (mcapi_test(&r2, &tSize,
&err))) {
         // KEEP WAITING
   }

   // allocate shared memory and send the ptr to TPU task
   sMem = shmemget(32);

   tmp_endpt = mcapi_endpoint_create(MCAPI_PORT_ANY, &err);
   CHECK_STATUS(err);

   mcapi_msg_send(tmp_endpt, tpu_remote_endpt, sMem, SHMEM_SIZE, priority,
&err);
   CHECK_STATUS(err);

   // connect the channels
   mcapi_sclchan_connect_i(tpu_endpt, tpu_remote_endpt, &r1, &err);
   CHECK_STATUS(err);

   mcapi_pktchan_connect_i(sig_endpt, sig_remote_endpt, &r2, &err);
   CHECK_STATUS(err);

   // wait on the connections
   while(!((mcapi_test(&r1, &tSize, &err))) && (mcapi_test(&r2, &tSize,
&err))) {
        // KEEP WAITING
   }

   // now open the channels
   mcapi_sclchan_recv_open_i(&tpu_chan, tpu_endpt, &r1, &err);
   CHECK_STATUS(err);

   mcapi_pktchan_recv_open_i(&sig_chan, sig_endpt, &r2, &err);
   CHECK_STATUS(err);

   // wait on the channels
   while(!((mcapi_test(&r1, &tSize, &err))) && (mcapi_test(&r2, &tSize,
&err))) {
        // KEEP WAITING
   }
```