

```

////////////////////////////////////
// The SIG Processing Task
////////////////////////////////////
void SIG_task() {
    mcapi_endpoint_t cntrl_endpt, cntrl_rmem_endpt;
    mcapi_pktchan_send_hndl_t cntrl_chan;
    mcapi_request_t rl;
    mcapi_status_t err;

    // init the system
    mcapi_initialize(SIG_NODE, &err);
    CHECK_STATUS(err);

    cntrl_endpt =
        mcapi_create_endpoint(SIG_PORT_CNTRL, &err);
    CHECK_STATUS(err);

    mcapi_get_endpoint_i(CNTRL_NODE, CNTRL_PORT_SIG,
        &cntrl_rmem_endpt, &rl, &err);
    CHECK_STATUS(err);

    // wait on the rmem endpoint
    mcapi_wait(&rl, NULL, &err);
    CHECK_STATUS(err);

    // NOTE - connection handled by control task
    // open the channel
    mcapi_open_pktchan_send_i(&cntrl_chan, cntrl_endpt,
        &rl, &err);
    CHECK_STATUS(err);

    // wait on the open
    mcapi_wait(&rl, NULL, &err);
    CHECK_STATUS(err);

    // All bootstrap is finished, now begin processing
    while (1) {
        // Read sensor & process signal
        struct SIG_DATA sDat; // populate this with results

        // send the data to the control process
        mcapi_pktchan_send(cntrl_port, &sDat, sizeof(sDat),
            &err);
        CHECK_STATUS(err);
    }
}

```