

XMesh Advanced Features

This section shows the features of XMesh including how to use the XMesh end-to-end transport service and how to send downstream commands from the base station to the motes. This software requires a Windows PC with MoteWorks, two motes, and one gateway board. In the MyApp subdirectory /lesson5 it shows how to use XMesh end-to-end acknowledgment, which have code to modify transport request to the base station. There is also a yellow LED light that blinks when a message is received. The MyApp will be installed onto two motes, one of the motes will be the sensor node while the other one will function as the base station. The mote you wish to use as the sensor node should be plugged into the programming board and you must install the MyApp to the node. The red and green lights will flash until a network is formed, once the network is formed the yellow light will flash. The MyApp will now have the ReceiveAck file which allows for interface writing. The ReceiveAck requires a callback function that is generated by XMesh. MODE_UPSTREAM_ACK tells XMesh to send a message acknowledging that the message was received to the base station. ReceiveAck.receive is another acknowledgment message that confirms a message has arrived from the base station and the LED light will flash green. To ensure the reliability of the message transport, there is an option MODE_UPSTREAM_ACK, which re-sends the message after an allotted period of time to ensure the message, is received. The MyApp subdirectory /lesson6 shows how to implement command processing. This application requires 2 motes, one will function as the sensor node, and the other as the base station, which is plugged into the programming board and connected to your PC. To send commands to the mote you must have XServeterm running on the computer to configure the sensor network. This will be used to send commands to the sensor nodes to show the downstream command capabilities. The first command to try is get_config, which will return the current configuration parameters for a mote. To get the configuration parameters for the other node simply type get_config 1. This shows that you have communication with the nodes. The Next command is set_rate and is used to change the motes sampling rate. XCommandC component of MyApp provides the functionality for processing downstream commands, the XCommand provides a single event name received which implements the application module and is signaled when a command arrives to the node.

Data Logging Application

This section teaches you how to read and write data from external flash on a mote. This software requires a Windows PC with MoteWorks, two motes, and one gateway board. The component that allows the user to read and write operations at the external flash is ByteEEPROM. This allows you to log the number of light sensor readings in the external flash. When a new reading comes it over-writes the previous reading. Once the new reading is written to the external flash the logged data is read back from the flash and is placed in a data packet on the computer. The node that has a node id of 0 will always be the base station. This node must be plugged into the programming board and when the LEF flashes green that means

the data packet has been sent to the UART and can now run XServe to display the incoming packets on the computer. Once XServe is running on your PC you can view the contents of the data packets being sent. The ByteEEPROM component is required to request memory in the external flash and carry out read and write operations. While the ByteEEPROM is being initialized the AllocationReq must be sent to the flash to request a byte section of the flash. All changes that need to be made use the interface AllocationReq, ReadData, and WriteData of ByteEEPROM. These are examples of how ByteEEPROM are used to access a mote's external flash for data logging. Although the mote's internal flash is not sufficient to fulfill data logging requirements in the real world this is a simulation of what occurs. This uses up the mote's battery power and should be used at the users discretion.

Source: MoteWorks Manual