Cooperative Mining Robots Testing Document

ISSUE RECORD				
Engineer	**Issue No.**	**Reason for Reissue**	**Page/Pages**	**Date**
Brooks Barrett Joe Galyean Tom Gauntner Shaun Heilmann Rob Lane Salman Mujahid John Nawrocki Kristen Radecsky Jason Salisbury Manon Skrzypecki Jeff Stofanak Andrew Thomas	1.00	New This document provides the test plans for each functional block diagram.	All	March 3, 2004

Table of Contents

Supervisory Station Functional Block Test Plan

Individual State Testing:

- (Note: At reset each port is configured input, logic zero.) After Initialize state is finished, set a pin as output and logic high to drive an 'Initialized LED'.
- A Startup Message is displayed to indicate proper communication to the PC.
- Require user acknowledgement of the Startup Message to indicate proper communication from the PC.
- Display a command prompt on PC when idle.
- Proper operation of all User I/O states will be tested/verified by their normal text based interaction with the user.
- Reception of data on the network will be indicated by an activity LED.
- Transmission of data on the network will be indicated by an activity LED.
- The activity of the 6 network/robot interaction states will be indicated by activity LEDs.
- The system shall have debugging levels that enables certain descriptors embedded throughout the system that are printed to the console for live examination of system operation and flow.

Hardware design conside rations for testing:

- Provide headers/Probe access points on the Tx/Rx of both USARTs.
- Provide headers/Probe access points on the RS232 Tx/Rx pins to the PC.
- Provide headers/Probe access points on the Differential Line Driver's A and B.
- Provide a bank of LEDs for indicating state/debug info.
- Provide a power on LED.
- Provide header/Probe access points for dirty and clean power supply.

Wired Communication Functional Block Test Plan

Individual State Testing:

- Each method within the program shall have a testing design associated with it.
- If a method should encounter an error an error specific 8-bit pattern will be generated.
- This 8-bit pattern will then be stored in memory for an external dump.
- Each method will be thoroughly tested before implementation.
- Each method shall pass on normal conditions and at the boundaries before the method is considered complete.
- If a method is revised during the course of implementation, these same tests will be run again to ensure proper operation.

Hardware design considerations for testing:

- All pins must be accessible for testing purposes.
- PORTA of the Atmega 162 shall provide the output for 8 LEDs.
- These LEDs will serve as visual status confirmation of the operation of the node.
- Two pushbuttons will also be provided to queue through the error status memory upon an error. This will ensure that the error is properly documented and taken care of.

Wireless Communication Functional Block Test Plan

Wireless Only:

- Send a byte from one node to the next
- Continually ping from one node back to the source
- Send a 50byte message to the node
- Receive message confirmation ping
- Continually send 50byte message (testing transfer speeds)
- Receive from multiple sources
- Determine which node to communicate with (ie continue transmitting to the same node even when another is within range)
- Change nodes but continue transmitting messages
- Re-transmit a message sent between nodes
- Specific errors will generate specific 8-bit patters which can be displayed or sent for debugging info.
- Test message handling at the receivers 45° range and ensure that the 10foot radius is reached.

Wireless/Wired Testing:

- Is a message sent from the robot relayed back down to another robot

Wireless/Navigation/Central Control:

- Is the message received ping returned from Axiom Board
- Does the Axiom Board receive a message

Navigation Functional Block Test Plan

Individual State Testing:

- Communication from central control is tested for receiving map information and mine work orders by comparing the information sent from central control and the information received by navigation.

- Communication to central control is tested for sending error and debugging information by comparing the information sent from navigation and the information received by central control.

- Communication with wireless is tested for receiving commands from wireless by comparing the information sent through wireless and that received by navigation.

- Communication with wireless is tested for sending commands from wireless by comparing the information sent by navigation and that received by wireless.

- Communication with motor control is tested for receiving task information from motor control by comparing the information sent by motor control and that received by navigation.

- Communication with motor control is tested for sending task information and commands from navigation by comparing the information sent by motor control and that received by navigation.

- The movement commands generated for robot motion are tested for collisions and shortest path through software testing for boundary conditions and scenarios involving other robots.

Hardware Testing:

- Test valid input, output connection with central control

- Test valid input, output connection with motor control

- Test valid input, output connection with wireless

Motor Control Functional Block Test Plan

Individual State Testing:

- Set the robot to Straight mode and provide a course made out of tape on the floor. Watch the robot navigate the course and make adjustments to provide the smoothest tracking possible. Test how tight of a corner the robot can make.
    - Test that each sensor reacts to dark and light the way it should and that the proper bits are transmitted for each case. Test that the response to each bit pattern from the sensors provides the right feedback to the PIC which produces the proper change in motor speed(PWM).
- Set the robot to Reverse mode and provide a course made out of tape on the floor. Watch the robot navigate the course and make adjustments to provide the smoothest tracking possible. Test how tight of a corner the robot can make. The robot should move very similar in reverse and straight modes.
- Set the robot to 180^{o} Turn mode and provide a straight line made out of tape on the floor. Watch the robot navigate the turn and make adjustments to make sure that the robot makes the turn and finds the path of tape every time.
- Set the robot to Right Turn mode and provide a Right Turn made out of tape on the floor. Watch the robot navigate the turn and make adjustments to make sure that the robot makes the turn and finds the new path of tape every time. Test different angles of paths to turn onto. Test how tight of a turn the robot can make.
- Set the robot to Left Turn mode and provide a Left Turn made out of tape on the floor. Watch the robot navigate the turn and make adjustments to make sure that the robot makes the turn and finds the new path of tape every time. Test different angles of paths to turn onto. Test how tight of a turn the robot can make.
- Test that when the robot reaches a tick mark it recognizes it and reports back to Navigation.
- Test that Motor Control can receive each of the five commands from Navigation.
- Test that when the panic sensor is activated, the power to each motor is interrupted until Navigation applies power again.

Central Control Functional Block Test Plan

General Testing:

- The system shall have debugging levels (chosen by external jumpers) that enables certain descriptors embedded throughout the system that are printed to the console for live examination of system operation and flow.
- Reception of data on the network will be indicated by an activity LED.
- Transmission of data on the network will be indicated by an activity LED.
- If power-on and subsystem check is OK, indicated with an INITed LED (Note: A Power-on/Reset automatically turns INIT LED off).
- If any failure is detected, indicated with a Failure LED if can.

Development Testing Procedure:

- Develop sanity interaction code to always be running.
- Develop network protocol code independently. Verify with byte loopbacks and direct Robot-Robot communications using predetermined packet swapping, with verification indicated on debug LEDs.
- After network protocol code is verified, parallel with Debug Console Display code and check Debug Level Jumper (DLJ) activation of the code.
- After network protocol and Debug Console Display (DCD) is verified, use Console to simulate ethernet network.
- Develop startup sequence, utilizing DCD and DLJ for reporting and verification of proper operation.
- Develop Poll Subsystems of main loop, use DCD and DLJ.
- Develop Error handling system, use DCD and DLJ.
- Develop supervisory interaction system, use DCD and DLJ.
- Develop Debug Supervisory Display (DSD) for debugging, testing.
- Develop check Shift Status system, use DCD and DLJ.
- Develop Navigation interaction, use DCD and DLJ
- Test proper operation of full system, utilizing DLJ, DSD, and DCD as necessary.

Hardware design considerations for testing:

- Provide headers/Probe access points on the Tx/Rx of the USART.
- Provide header/Probe access point on the Power Monitoring Flag.
- Provide headers/Probe access points on the Sanity Links.
- Provide a bank of 8 LEDs for indicating current functional block being executed.
- Provide two selector jumpers for choosing Debug Level
    - 00: No debug
    - 01: Display current function on 8 LEDs
    - 10: Activate Supervisory display of debug descriptors
    - 11: Activate ASCII/Console display with full debug descriptors
- Provide a power on LED.
- Provide header/Probe access points for dirty and clean power supply.

Sanity Functional Block Test Plan

Individual State Testing:

- At startup, the system is waiting for its clock cycle. The length of this first clock is to be tested via the scope. It should be 8ms long. After the end of the clock cycle, reset should be low.
- Test that reset remains low for 10 consecutive good handshakes.
- Test that reset remains low for 5 good shakes, followed by 1 bad shake, followed by 5 good shakes.
- Test that reset remains low for 5 good shakes, followed by 2 bad shake, followed by 5 good shakes.
- Test that becomes high after 3 bad shake, which are to come after 5 good shakes.
- Test that S0 = S1 for 3 consecutive cycles causes a reset.
- Monitor t such that it toggles as long as there is good handshaking.
- Test that t does not toggle in response to 1 bad handshake.
- Test that t does not toggle in response to 2 bad handshake.
- Test that t does not toggle in response to 3 bad handshake.
- Test that the red LED lights when there is a reset.

Hardware Design Considerations for Testing:

- Be sure that there is a valid connection to Central Control.
- Provide a red LED such that it lights when there is a reset.

Power Functional Block Test Plan

Individual State and Hardware Testing:

- The battery level alarm is tested for the low battery threshold.
- Communication with central control is tested for the battery level alarm being sent from power and received by central control.
- Emergency power switching mechanism is tested for it being turned on when the low battery threshold is reached.
- Power Distribution/ DC to DC converters (and corresponding circuits if applicable) to all modules are tested for the desired power requirements.
- Current overdraw circuit is tested for the current threshold.
- Communication with central control is tested for the current overdraw alarm being sent from power and received by central control.
- Main circuit breaker is tested for the desired current threshold.
- Battery charger switch is tested for its functionality by getting desired switch between charging and regular modes.
- Motor Control circuit breaker is tested for the desired current threshold.
- Central Control/ Navigation circuit breaker is tested for the desired current threshold.