Integration Plans for Lego Mindstorms NXT and MS Robotics Studio

Milestone IV – Project 3
Dennis O'Flaherty

Section 1: Introduction

Revision History:

Purpose and Scope: The purpose of this document is to describe the subsystems of the project and the procedures for integrating each subsystem into the whole system. It will briefly describe the role and development of each module and then discuss at a greater length, each module's future.

References: Reference documents can be viewed on the Microsoft Developers website as well as the Lego NXT website.

Section 2: Integration Strategy

There are two different kinds of modules developed for this project: PC-side and Brick-side. There are three PC modules, which are those software components with which the user directly interacts, namely the manual and autonomous control programs and the visual mapping component. Brick-side modules are components that have been encapsulated from the user's perspective: the actual hardware configuration of the robot. There are three Brick-side modules: the touch sensor, the movement subsystem and the ultrasonic sensor subsystem.

Prior to integration, each module must be tested thoroughly by the developer himself and then demonstrated to the other members of the group for approval. Once all
members of the group have approved that component, the actual integration testing is done. The individual steps for the testing from that point forward depends on the module itself.

The strategy for integration testing is functional grouping. This was selected because of the two distinct module types and the existence of a simple interface between the two functional groups. So far the Brick-side components have been integrated. The drive system started out as a 4-wheel design. When tested with a stub in Robotics Studio that simply made the robot go forward, backward and turn, it was modified to a 3-wheel design. The touch sensor was built and tested with the same stub and some obstacles and modified as necessary. The same approach was followed for testing the ultrasonic sensor module. The two components that now need to be integrated do not depend on each other so they can be tested separately. They only depend on the robot itself and the components of that have been sufficiently tested. Both control modes will be tested by running the robot through a few test courses. The obstacles will be selected to test how precisely the robot moves whether controlled by a human or a computer.

![Diagram](image-url)
Section 3: Program Stubs and Test Data

The software stubs used to test the hardware configuration were very simple. They used no logic and only told the robot to turn or move backwards or forwards. These proved useful in refining the design of the hardware. Gradually, these stubs were increased functionally and became the foundation for the autonomous navigation software component. A few sample obstacles were used to test the touch sensor and ultrasonic sensor. The code for the control modes will be analyzed once written and edge cases will be determined to evaluate the interaction between those components. The data used to test the software modules will be more than just stubs. Obstacle courses will be designed to see how effectively the robot can navigate, whether it is running the autonomous control mode or being controlled by a human. Extreme cases will be tested, such as an empty room and an extremely crowded room just to see how it reacts, but the emphasis will be placed on realistic environments to make sure that both control modes work well and are useful. The software will be revised as needed to adjust to an assortment of reasonable rooms; the room will not be set up to adapt to the software

Section 4: Responsibilities and Schedule

James has been responsible for the hardware modules thus far and has done the integration testing for them in the hardware functional group. He will continue participating as a consultant for the other three members as they develop software for the hardware assembled by him. Karl is developing the visualization software component. This will utilize the touch sensor and the ultrasonic sensor components to map out a sample room and display it visually on a computer screen. Dennis and Dave are refining the automatic control component and
composing a manual control software component. The autonomous mode component was partially developed by James during the integration testing of the hardware modules but still needs to be expanded in function and refined to utilize the hardware capabilities instead of testing them. The manual control mode will depend almost exclusively on the wheel module but may need some input from the touch sensor so a user cannot easily damage the robot by repeatedly running it into a wall. The automatic control mode will depend on all of the modules. The wheel component will control motion. The ultrasonic and touch sensors will be necessary for obstacle avoidance and navigation. Since there are currently two identical robots, the tests conducted by Karl and Dennis and Dave can be done simultaneously, allowing everyone to complete their tasks more quickly.

Section 5: Problem Recording and Resolution

Components will only be reworked if it fails when being integrated with the rest of the system. In general, thoroughly tested lower order modules (namely the hardware) will be assumed to work and the software modules will be designed to adapt to them. Unless the software reveals a major defect in the hardware, the hardware will not be designed to adapt to the software. For instance, a software stub during testing of the wheels module revealed that a 4-wheel configuration caused the front wheels to slide while turning and the hardware was revised because it was a physical problem. When the latency of sending commands over Bluetooth was discovered to present a small problem, the software was adapted to that instead of revising the hardware.

Because the work is evenly divided among the team members and no more modules have been queued up, suspension criteria are very high. A module has to fail
completely in order to be suspended because the only dependencies for incomplete modules have already been thoroughly tested. The only modules that have not been tested for integration are the software components which do not depend on each other, only the hardware. With two robots, this allows two separate integration tests to be run simultaneously. This allows us to find and fix problems more quickly since it reduces module dependency by increasing the resources.