Harris Search & Rescue Robot

Team Members

Tyler Culp tculp2012@my.fit.edu
Milton Stafford mstafford2012@my.fit.edu
Devin Martinez dmartinez2012@my.fit.edu
Faculty Sponsor: Philip Chan pkc@cs.fit.edu
Client: Harris Corporation

Preliminary-level Design Review (PDR)

The Preliminary-level Design Review was the first formal presentation to our client, where our team outlined in detail the logical design of the system. This review included an overview of previously created documentation and additions such as configuration items for hardware and software, as well as trade studies. The review resulted in a 'go ahead' from our client, and we are on track to present a formal Critical Design Review (CDR) on February 29.

Current Milestone Tasks

Task	Completion %	Tyler	Milton	Devin	To do
Create Simulation Env't	50%	70%	15%	15%	Add more tests, implement additional aspects, such as friction and power
Define Soft. Config Items	100%	70%	15%	15%	
Identify Software Interfaces	100%	70%	15%	15%	
Identify Software/Hardware Interfaces	100%	15%	70%	15%	
Software Block Diagrams	100%	30%	30%	40%	
Define Software Requirements	50%	0%	50%	0%	For each: add source requirement, add detail in requirement

Create Simulation Environment

The team was tasked with creation of a simulation environment for the robot. After much consideration, the team decided on a simple design for simulation instead of creating an entire simulated rubble pile. The team would create obstacles based off of the requirements given in the initial A-spec. This way, the requirements would be easy to test with proposed designs. If the simulated robot failed the obstacle, the real robot would most likely fail to meet the requirement.

Define Software Configuration Items

The software configuration items have been identified as modules that the software will have, based on separate, distinct functionality. The modules are split into two main sections, the Workstation and the Robot. The workstation section primarily handles input and output to and from the operator, and communicates with the robot. The robot section handles the movement of the robot, as well as the input from the sensors, outputting the operator's voice, and communicating with the workstation.

Identify Software Interfaces

We have identified the interfaces that will exist between the software modules. For example, the Controller module will send the input data to the Admin for the Workstation, which will send the data to the Networking module. The data will get transmitted to the Networking module on the robot, from there to the robot Admin, and finally to the Locomotion module. Thus there is an interface from the Controller to the Admin_W, from the Admin_W to Networking_W, from Networking_W to Networking_R, etc.

Identify Software/Hardware Interfaces

We have also Identified the interfaces between the software modules and the hardware components. For example, the Controller module interacts with the input device (controller/ keyboard), the Audio_In modules interact with the microphones, the Video_Out interacts with the screen, etc. The existence of these interfaces have been identified, however the specific attributes and protocols have not been defined.

Software Block Diagrams

The teamed created block diagrams that illustrated the communication between each software configuration item. There are two main configuration item groups: robot and workstation, and both connect via a networking module. Both groups are managed by an

administrator that directly communicates with the respective networking module. Other modules on the robot's side include: SystemStats, Locomotion, GPS, AudioInput, AudioOutput, and PowerStates. Other modules on the workstation side include: Controller, VideoOutput, AudioInput, and AudioOutput.

Define Software Requirements

Once the team defined the function of each subsystem (e.g. Communication) the CS members defined the function of software modules necessary to achieve the overall goal. To illustrate this, a requirement of the robot is that it must receive instructions from the operator to control direction. Now, the software will provide the infrastructure to relay commands received via the transmitter onboard the robot to control the motors. So we've defined the requirement of each necessary module using a justification similar to the one above, such that we have requirements for our configuration items.

Team Members

Tyler Culp

Tyler worked on the simulation environment, the software configuration items, and the software internal interfaces. The SWCIs and the software interfaces will be used to determine the overall structure of the program. Additionally, he has assisted with the block diagrams and identifying the software/hardware interfaces.

Devin Martinez

Devin worked on adding to and organizing initial block diagram sketches; more specifically, the block diagrams illustrating the software configuration items. He helped with defining the modules that go in both CI groups and how they communicate with each other.

Milton Stafford

Milton worked on the block diagrams, especially the hardware/software interface diagram that identifies the interface between the hardware components and the software modules, as well as actually identifying the hardware components with which each module will interact.

Milestone 4 Tasks

Task	Tyler	Milton	Devin
Define Software Interfaces	33%	33%	33%
Define Software/Hardware Interfaces	25%	25%	50%
Define Software Dependencies	25%	25%	50%
Revise Software Requirements	15%	70%	15%
Improve Simulation Env't	70%	15%	15%

Define Software Interfaces

We will define the attributes and protocols of the interfaces between the modules. Several of the interfaces, such as the audio transmission, will be forced to match existing data streaming protocols. Other interfaces, such as that between and Controller and Admin modules, will be more flexible and may use JSON strings or some other plain text format.

Define Software/Hardware Interfaces

Now that the interface between modules and components has been identified, it needs to be described in more detail, outlining exactly what is "flowing" across the interface, and how it flows. In the software case, this means choosing IO signatures, protocols, and APIs.

Define Software Dependencies

After PDR, it was agreed upon that the team would need to determine what hardware needs to be functioning correctly in order to test software modules relevant to that piece of hardware. Additionally, some hardware will come with software or will require drivers for our software to interact with it.

Revise Software Requirements

Based on the feedback received from PDR, we know that the Software Requirements need to be updated to include source requirements from the A-Spec, and use language consistent with the "shall" statements included in the A-Spec. The requirements also need to be more specific.

Improve Simulation Environment

More tests will be added to the simulation, in order to see if proposed designs will meet a greater amount of requirements. Additionally, we plan on adding additional features, such as tracking expected power consumption, taking friction into account, modeling the movement of the treads themselves, etc.

Sponsor Feedback

Faculty Sponsor Approval

"I have discussed with the team and approve this project plan. I will evaluate the progress and assign a grade for each of the three milestones."

Signature: _____ Date: _____

Sponsor Evaluation

Tyler Culp	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Milton Stafford	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Devin Martinez	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

Signature: _____ Date: _____