AR:IN

Autonomous Robotics: Intersection Navigation

GROUP 11

Development Process and Implementation

Derek Arts

Jonathan Klawunn

Kate Keskikyla

Kerianne Rikley

Nabil Hamza

Trent Stevenson

Development Process and Implementation

Project goals and requirements

Development Process workflow:

Design Phase

Completed Design Document

Design Documentation Review

Hardware Design Specifications

Software Design Specifications

Software Requirements Document given hardware design limitations

Mechanical Requirements Document

Mechanical Design Specifications

Mechanical, Hardware and Software Requirements

Hardware Requirements Document given mechanical design limitations

Implementation/Testing Phase

Hardware Design Implementation

Mechanical Testing and Revision

Mechanical Design Implementation

Detailed Design Specifications Documents

Mechanical Prototype

Revised Mechanical Prototype

Hardware Prototype

Hardware Testing and Revision

Software Implementation

Revised Hardware + Mechanical System

Hardware/Mechanical Integration and testing

Revised Hardware Prototype

Software initial baseline

Final Presentation

Final working prototype

Final Testing and bug fixes

Final Prototype

Software Integration

Revised Software Prototype

SiL testing and revisions

Detailed Design Specifications Documents

Process Descriptions

Design Phase:

**Mechanical, Hardware and Software Requirements**

* Project goals and requirements will be translated into specific mechanical, hardware and software requirements

**Mechanical Design Specifications**

* Mechanical requirements document will be translated into design document

**Hardware Design Specifications**

* Hardware design requirements document will be translated into design document
* Limitations imposed by the mechanical design will be considered when designing hardware

**Software Design Specifications**

* Software design requirements document will be translated into design document
* Limitations imposed by the hardware design will be considered when designing software

**Design Documentation Review**

* All design documentation will be reviewed with TA or Dr.Wassyng
* Parts will be ordered

Implementation/Testing Phase:

**Mechanical Design Implementation**

* Mechanical Design will be implemented based on design document
* Will include making 2 cars that can move as well as all the mounting brackets needed for sensors and hardware
* Sub-systems will be tested while implementation takes place

**Mechanical Testing and Revision**

* Testing of complete mechanical system will take place
* This will include making sure the cars are able to move and that hardware can be mounted on them easily

**Hardware Design Implementation**

* Hardware Design will be implemented based on design document
* This will include connecting sensors to microcontroller and making sure it can read inputs as well as drive the motors

**Hardware Testing and Revision**

* Testing of hardware system will take place
* This will include making sure all sensors and motors are working by uploading test code to the microcontroller

**Hardware/Mechanical Integration and Testing**

* Hardware will be integrated with car and tested again to make sure hardware and mechanical components still work

**Software Implementation**

* Software design will be implemented based on design document
* This will include coding system in C

**SiL Testing and Revision**

* Software will be tested using SiL simulation as well as Simulink to validate code

**Software Integration**

* Software will be uploaded to hardware and tested to see if code runs system

**Final Testing and Bug Fixes**

* Final testing will take place on complete system to see if it meets requirements as well as the fixing of any last minute bugs

Acceptance Criteria for Outputs

**Mechanical Requirements Document**: A detailed document specifically explaining the requirements for the mechanical components of the system.

**Hardware Requirements Document**: A detailed document specifically explaining the requirements for the hardware components of the system.

**Software Requirements Document**: A detailed document specifically explaining the requirements for the software components of the system.

**Mechanical Prototype**: A mechanical prototype meeting all of the requirements of the Mechanical Requirements Document, will include cars plus all sensor mounting bracket

**Hardware Prototype**: A hardware prototype meeting all of the requirements of the Hardware Requirements Document, will include micro-controller as well as sensors and motors

**Software Prototype**: A Software prototype meeting all of the requirements of the Software Requirements Document, will include full software control system for facilitating 4-way intersection behaviour

**Revised Mechanical Prototype**: A mechanical prototype meeting all of the requirements of the Mechanical Requirements Document as well as any improvements made during the testing period.

**Revised Hardware Prototype**: A hardware prototype meeting all of the requirements of the Hardware Requirements Document as well as any improvements made during the testing period.

**Revised Hardware and Mechanical System**: A system composed of the revised mechanical system and revised hardware system integrated together, will include cars will all sensors and the microcontroller will be mounted.

**Revised Software Prototype**: A Software prototype meeting all of the requirements of the Software Requirements Document as well as any improvements made during the testing period.

**Revised Hardware, Mechanical, and Software System**: A system composed of the revised mechanical system, revised hardware system, and revised software system integrated together. This will include the cars with sensors and microcontroller, as well as functioning code.

Potential Tools and Resources

* McMaster Machine Shop
* Matlab and Simulink
* Code Comparison Software ex. Beyond Compare
* GitHub
* Eclipse CDT

Coding Standards

* Software will be divided up into sub-systems
* Each sub-system will have top module that communicates with the other sub-systems but all other modules within each sub-system will be hidden to the other sub-systems

Version Control

For mechanical and hardware systems version testing will not likely be needed unless a change is made to the initial design. The software will have baseline cases to insure that the current code meets requirements and so that there is always a proper working version of the software as a safety net and the progress in the software can be tracked. GitHub will be used to upload baseline versions of the code and there will also be a section to push bug fixes.

Bug Tracking

When a bug is found, a software change will be requested and documented in a Microsoft Excel spreadsheet in a Google Document. The person who found the bug and the person who wrote the code will then review the code to see if they can find the source of the bug. Once the source is found, it will be corrected and then the code will be check again to make sure that no new bugs have occurred and the old bug is fixed.

Distribution of Work

Work will be assigned based on strengths of group members