

<b>Project Title:</b>	Embedded Web Server Automation
<b>Organization:</b>	Harris RF Communications
<b>Primary Contact:</b>	Nathan Ransom
<b>Email:</b>	<a href="mailto:nransom@harris.com">nransom@harris.com</a>
<b>Phone:</b>	(585) 242-3908

***THIS INFORMATION IS EAR CONTROLLED***

*THIS INFORMATION IS CONTROLLED BY THE U.S. DEPARTMENT OF COMMERCE EXPORT ADMINISTRATION REGULATIONS 15 CFR 730-774, ECCN EAR99. EXPORT OR DISCLOSURE TO FOREIGN PERSONS MAY VIOLATE U.S. FEDERAL REGULATIONS*

**Background Information**

Harris RF Communications is a division of Harris Corporation that designs, tests, and manufactures military radios for both the U.S. government and international markets. These military radios span a range of sizes from small handheld radios to larger man portable (manpack) radios to much larger systems installed in strategic ground or on-board ship facilities. Many of the radio products are constrained by size requirements which results in limited user interface and LCD display capabilities when compared with commercial products like PDA's or PC's.

The availability of increased processing power and network connectivity available to embedded computing platforms has dramatically increased the desire and need for embedded devices to be able to be controlled and configured through network interfaces. In particular, embedded web servers have offered up an extremely powerful method for exposing these interfaces in platform independent ways in a variety of different devices from advanced computing devices to commercial wireless routers. The problem is that embedded devices tend to differ greatly in the types of configuration available and this requires that engineering resources are spent repeating the same type of work for a variety of different embedded platforms. The goal of this project will be to lessen this burden on software teams by automating the creation of configuration and REST resources through the use of code generation.

**Project Description**

This project will build upon the open source embedded webserver Mongoose (<https://code.google.com/p/mongoose/>). The Mongoose server has been ported to a variety of different platforms and supports many different features common to more advanced web platforms including SSL, WebSockets, etc. Mongoose provides an extremely flexible and light-weight server for inclusion into embedded platforms.

The goal of this project will be the creation of a tool that allows a developer to specify the desired configuration of an embedded system through an XML file. The tool will then generate

the necessary code for interfacing to the Mongoose webserver, HTML pages for presenting a UI to the user, as well as any necessary routing and handling of a JSON-based REST API for the configuration. The end goal of the generated code would be that a developer using the output of the tool would only need to implement a simple API to be notified of configuration changes.

The second goal of this project includes the ability for custom functionality to be included in the REST API that isn't necessarily tied to the configuration. The code generation tool would be required to configure Mongoose to allow for custom methods such that arguments could be passed to the API through JSON parameters and the end developer could use this mechanism for RPC functionality into the embedded device. The team will be required to demonstrate this functionality through the manipulation of hardware on the embedded platform (i.e. toggling an LED, sending data over serial link, etc).

A few stretch goals are included with the project as well. The first stretch goal is to generate documentation about all the interfaces that are created including such details as all valid REST API's, parameters, and available configuration. The second stretch goal will be generated code that validates the parameters presented through the different APIs. The sponsor envisions this could take several forms including Javascript validation on the HTML pages, limiting valid inputs through the use of drop-downs, as well as server side validation for the submissions. Additional features suggested by the team or advisor are also welcomed and encouraged.

## **Project Scope**

The project team will be expected to handle gathering of requirements, design and implementation of the code generation tool. If the team desires to do so, the sponsor welcomes input and feedback into some of the details of the project or additional features that they feel would provide value.

## **Project Challenges**

The main challenge of this project will likely be integrating with the webserver in a method that is easy to generate source code. While this task is not itself extremely difficult, it is a different method of approaching the design of the code in order to make the code generator and end user friendly.

## **Constraints & Assumptions**

- Integrate with the C-based Mongoose server
- End user programming interfaces defined and implemented using C++
- Code generation tool in a platform independent language such as Ruby or Python
  - The sponsor has utilized Ruby for similar tasks in the past and would recommend it be considered
- Documentation provided in Adobe PDF format
- Use CSS formatting for the user presented HTML pages to allow for simpler formatting changes and customization

- The custom webservice integration and REST API will operate on the sponsor provided hardware platforms.

## **Sponsor-Provided Hardware and Software**

The Mongoose webservice is an open source project available from <https://code.google.com/p/mongoose/>. The sponsor will provide embedded platforms (Raspberry PI) for demonstration of the output of the code generation tool. These platforms are not expected to be returned to the sponsor and it is expected that they will remain with team members or become property of the RIT Software Engineering department upon conclusion of the project. The details of this will be worked out at a future time between the sponsor and the Software Engineering Department.

## **Project Search Keywords**

Embedded, Web, REST, JSON, Mongoose

## **Department of Software Engineering Required Deliverables**

1. Project website holding all work products and project artifacts maintained in the project account on the se.rit.edu web server.
2. Project plan, schedule and process methodology definition prepared by the end of week 3 of the first term.
3. Tracking report for time/effort worked on the project, and at least two other product/process metrics appropriate to the project and development methodology. Tracking reports updated on the project website at least every two weeks.
4. Interim status and final project presentations
5. Project poster and presentation at “SE Senior Project Day”
6. Project technical report

## **Sponsor and Project Specific Deliverables**

1. Requirements document
2. High Level Design document
3. Detailed Design document
4. User Manual (including section on building source if necessary)
5. Software package, including source code that can be built

## **Proprietary Information**

This project does not any proprietary information.

## Project Agreements and Assignment of Rights

RIT policy gives students full ownership of any work done as part of coursework which includes their work on senior project. As the sponsor of a course project, you can select one of three approaches for dealing with ownership of project artifacts and intellectual property, and the disclosure of proprietary information. If you seek assignment of rights, the individual team members will sign a project agreement based on the rights that you want.

If you are requesting that the team assign project rights to you, please get any corporate and legal clearances that you feel are needed to use the **unmodified** project agreement, before submitting your project proposal. Indicate that this has been done with an **X** in the left box below.

<u><b>X</b></u>	<b>Corporate and Legal Clearance of Project Agreement</b>
	This project will either be an Open Source Project not requiring assignment of rights, or we have the necessary approvals to use the <b>unmodified</b> project agreement. (Note: The project agreements are cleared for RIT internal projects.)

Select one of the following approaches for assignment of the project artifacts and intellectual property, and the disclosure of proprietary information by placing an **X** in the box to the left of the appropriate paragraph below.

_____	<b>Assignment of Full Rights</b>
	If a team is assigned to this project, all students on the team will sign a standard Student Course Project Intellectual Property and Non-Disclosure Agreement. This agreement assigns the rights to the team's project work to the sponsor, and describes the process whereby the project sponsor can reveal proprietary information to the team. For non-RIT projects, the faculty coach will sign a standard Faculty Course Project Non-Disclosure Agreement which describes the same process for revealing proprietary information.
_____	<b>Assignment of Limited Use Rights</b>
	If a team is assigned to this project, all students on the team will sign a standard Student Course Project Limited Use and Non-Disclosure Agreement. This agreement assigns the sponsor rights to the team's project work for internal or non-commercial use by the sponsor. The sponsor may maintain and extend the project but not transfer it to a third party or use it in a commercial product. The project team will retain patent and commercialization rights. The agreement also describes the process whereby the project sponsor can reveal proprietary information to the team. For non-RIT projects, the faculty coach will sign a standard Faculty Course Project Non-Disclosure Agreement which describes the same process for revealing proprietary information.
<u><b>X</b></u>	<b>Open Source Project</b>
	If a team is assigned to this project, the team will develop this as an open source project and will publish all artifacts via an open source mechanism agreed upon through discussions with the project sponsor. The sponsor will gain access to project artifacts only through this open source repository. No rights need to be assigned exclusively to

the project sponsor, and there will be no transfer of proprietary information.
--

The agreements and policies can be found at:

- Student Course Project Intellectual Property and Non-Disclosure Agreement  
<http://www.se.rit.edu/~se561/CourseInformation/StudentCourseProjectAgreement.doc>
- Student Course Project Limited Use and Non-Disclosure Agreement  
<http://www.se.rit.edu/~se561/CourseInformation/StudentCourseProjectLimitedAgreement.doc>
- Faculty Course Project Non-Disclosure Agreement  
<http://www.se.rit.edu/~se561/CourseInformation/FacultyCourseProjectAgreement.doc>
- RIT Intellectual Property Policy C3.0. The project agreements are consistent with section C3.0(5)(B)(2)  
<http://www.rit.edu/academicaffairs/policiesmanual/sectionC/C3.html>