|  |  |  |
| --- | --- | --- |
| nscl.PNG | National Superconducting Cyclotron Laboratory | msu.png |

Preliminary Project Plan

|  |  |
| --- | --- |
|  |  |
| Project Name | CA Protocol on Rabbit |
| Project Code | CAPOR |
| Account | 2306-001-EE-014 (Work Order for Estimation) |
| Department | EE |
| Project Leader | John Priller |
| Project Coordinator | Vasu Vuppala |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Signature | Date |
| Prepared By | Vasu Vuppala |  |  |
| Reviewed By | John Priller |  |  |
| Approved By | John Vincent |  |  |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 1.0 | 1/20/10 | Vasu V | Initial |
| 1.1 | 2/4/10 | Vasu V | Added cost, approval form. |
| 1.2 | 2/5/10 | Vasu V | Modified the approval form |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
| **Preliminary Project Plan Approval Form** | |
| **Approval Procedure*:***   1. Project Coordinator (PC) sends the Preliminary Project Plan (PPP) to Department Head (DH) for approval. 2. DH approves the PPP by signing this form or acknowledging it via written communication such as email 3. PC sends the PPP to the Customer 4. Customer approves the PPP by signing this form or acknowledging it via written communication such as email, and submits a Work Order Request 5. DH routes the Work Order to the Project Leader, who assigns the team and effort to the Work Order 6. The project team starts work on the project, and records its efforts against the Work Order | |
| ***Project***: CA Protocol on Rabbit (CAPOR) | ***Customer***: Kelly Davidson |
| *Project Leader (PL)*: John Priller | *Project Coordinator (PC):* Vasu Vuppala |
| ***Project Description***:  Most of the embedded controllers used in the NSCL control system, developed by the Electrical Engineering Department (EE), are based on RabbitCore Module (RCM) from Rabbit Corporation. The primary mode of communication with the embedded controllers is through the ModBus/TCP protocol. The ModBus/TCP protocol has several limitations with respect to the NSCL control system. To address some of the limitations, this project will implement the EPICS Channel Access (CA) Protocol on the RCM platform. | |
| ***Estimates****:*   |  |  | | --- | --- | | *Effort*: 943 Person-Hours | *Cost*: | | *Start Date*: 08 Feb 2010 | *End Date*: 07 Aug 2010 | | |
| ***Project Team****:*  John Priller  Mark Davis  Vasu Vuppala | |
| ***Comments***: | |
| *Approvals*:   |  |  | | --- | --- | | Department Head: | Date: | | Customer: | Date: | | |

Table of Contents

[1 Project Overview 6](#_Toc253149168)

[1.1 Introduction 6](#_Toc253149169)

[1.2 Business Need 6](#_Toc253149170)

[1.3 Value Engineering 6](#_Toc253149171)

[2 Project Scope 7](#_Toc253149172)

[2.1 Scope Statement 7](#_Toc253149173)

[2.1.1 Scope Description 7](#_Toc253149174)

[2.1.2 Product Acceptance Criteria 7](#_Toc253149175)

[2.1.3 Project Deliverables 7](#_Toc253149176)

[2.1.4 Project Exclusions (Out of Scope) 7](#_Toc253149177)

[2.1.5 Project Constraints 8](#_Toc253149178)

[2.1.6 Project Assumptions 8](#_Toc253149179)

[2.2 Milestones 8](#_Toc253149180)

[2.3 Work Breakdown Structure 9](#_Toc253149181)

[2.3.1 WBS Dictionary 9](#_Toc253149182)

[3 Project Strategy 10](#_Toc253149183)

[3.1 Critical Success Factors 11](#_Toc253149184)

[3.2 Project Life Cycle 11](#_Toc253149185)

[3.3 Product Life Cycle 11](#_Toc253149186)

[4 Project Schedule 11](#_Toc253149187)

[4.1 Schedule Management 12](#_Toc253149188)

[5 Project Budget 12](#_Toc253149189)

[5.1 Cost Management 12](#_Toc253149190)

[6 Human Resource Management 12](#_Toc253149191)

[6.1 Organization 12](#_Toc253149192)

[6.2 Roles and Responsibilities 12](#_Toc253149193)

[6.2.1 Responsibility Assignment Matrix 12](#_Toc253149194)

[6.3 Resource Loading and Release Plan 13](#_Toc253149195)

[7 Quality Management 13](#_Toc253149196)

[7.1 Quality Metrics 13](#_Toc253149197)

[7.2 Quality Management Plan 13](#_Toc253149198)

[7.3 Quality Improvement Plan 13](#_Toc253149199)

[8 Risk Management 14](#_Toc253149200)

[8.1 Risk Breakdown Structure 14](#_Toc253149201)

[8.1.1 Risk Register 14](#_Toc253149202)

[9 Communications Management 14](#_Toc253149203)

[9.1 Stakeholder Register 14](#_Toc253149204)

[9.2 Communications Plan 14](#_Toc253149205)

[10 Procurement Management 14](#_Toc253149206)

[11 Configuration Management 15](#_Toc253149207)

[11.1 Configuration Items 15](#_Toc253149208)

[11.2 Change Control 15](#_Toc253149209)

[11.3 Change Classification (Major vs Minor) 15](#_Toc253149210)

[12 References 15](#_Toc253149211)

[13 Glossary 15](#_Toc253149212)

# Project Overview

## Introduction

National Superconducting Cyclotron Laboratory (NSCL) is a world leader in rare isotope research and nuclear science education. NSCL scientists and researchers conduct advanced research in fundamental nuclear science, nuclear astrophysics, and accelerator physics. To facilitate the research, NSCL operates multiple particle accelerators.

The control system at NSCL enables operators to control the various devices in the lab: ion sources, cyclotrons, beam lines, experimental devices, and auxiliary equipment. It is based on the Experimental Physics and Industrial Control Systems (EPICS) standard. EPICS is a distributed soft real-time control system for large scientific experiments.

Most of the embedded controllers used in the NSCL control system, developed by the Electrical Engineering Department (EE), are based on RabbitCore Module (RCM) from Rabbit Corporation. The primary mode of communication with the embedded controllers is through the ModBus/TCP protocol. The ModBus/TCP protocol has several limitations with respect to the NSCL control system. To address some of the limitations, this project will implement the EPICS Channel Access (CA) Protocol on the RCM platform.

## Business Need

The ModBus/TCP protocol is simple, and has limited requirements in terms of memory and compute cycles. Hence, it is a good match for the limited resources on the RCM platform. However, it does not have provisions for asynchronous event notifications, and has to be polled for device status. This puts a burden on the network. It does not provide logical naming for device data, which makes it cumbersome to maintain its configuration especially in an EPCIS-based environment.

Switching to the CA protocol will conserve network resources (asynchronous event notifications) and staffing resources (standard naming, native protocol).

## Value Engineering

* Native protocol for EPICS
* Will help with productivity
* Consumes less network bandwidth

# Project Scope

## Scope Statement

### Scope Description

This project will develop the CA protocol on the RCM platform. It may not be possible to implement the complete protocol due to the limited resources on RCM. In any case, it is not envisioned to have the embedded controller be a full-fledged EPICS IOC (Input/Output Controller). However, the following features must be implemented:

* PV Search
* Channel Creation
* Channel Read
* Channel Write
* Event Notification: PV Change
* Server Beacons

### Product Acceptance Criteria

The implementation on a RCM3200 (RCM with a Rabbit 3000 processor) must pass the benchmark test.

### Project Deliverables

The deliverables are:

* The Binary Image
* Source Code
* Installation Manual: Describes how the image should be deployed
* User Manual
* Design documentation
* Acceptance Test Benchmark
* Test Plans (optional)
* Recommendations, if the implementation is not feasible

### Project Exclusions (Out of Scope)

The RCM based controllers currently in operation at NSCL can be divided into four categories depending on their CPU (Rabbit 2000 and 3000) and RTOS (DEBROS or non-DEBROS). This project will implement and test CA protocol on only one category: Rabbit-3000 based RCM running DEBROS. Non-DEBROS or Rabbit-2000 based RCMs are out of the scope.

The following features of CA protocol are out of scope:

* Access Control
* Repeater

### Project Constraints

* The solution must be built on DEBROS.
* The solution must run on the current RCM modules in EE department
* It must not alter the current update mechanism for embedded software

### Project Assumptions

* This project will test the implementation for only one device (8-port Power Supply Controller)
* It may not be feasible to implement CA protocol on Rabbit. As and when this becomes apparent, the stakeholders will be notified, and the project will be terminated.

## Milestones

* Project Plan
* Architectural Design
* Low Level Design
* Version 1.0 b0
* Version 1.0 b1
* Version 1.0
* Project Closure Report

## Work Breakdown Structure

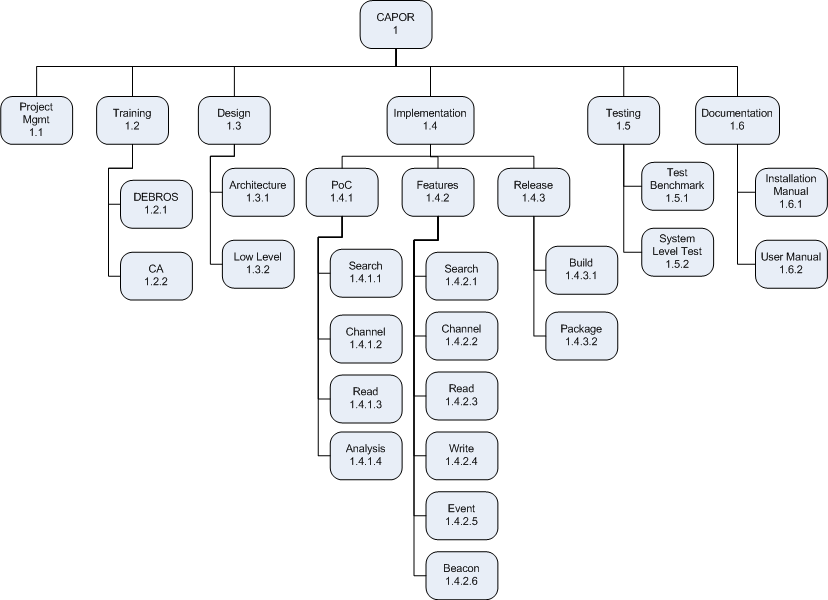


Figure Work Breakdown Structure

### WBS Dictionary

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Assigned To** | **Assign Date** | **Est Due Date** | **Est Cost** |
| 1 | CAPOR | CA Protocol on Rabbit |  |  |  |  |
| 1.1 | Project Management | Includes all activities of PM. |  |  |  |  |
| 1.2 | Training | Training required for implementing the project |  |  |  |  |
| 1.2.1 | DEBROS | Learning about DEBROS and documenting it |  |  |  |  |
| 1.2.2 | Channel Access | Understanding EPICS CA in detail |  |  |  |  |
| 1.3 | Design | Analysis & Design activities |  |  |  |  |
| 1.3.1 | Architecture | High-level design and the resulting document. |  |  |  |  |
| 1.3.2 | Low-Level | Detailed design |  |  |  |  |
| 1.4 | Implementation | This includes coding, refactoring, and unit test (CRUT) |  |  |  |  |
| 1.4.1 | Proof of Concept | Implementation of the POC |  |  |  |  |
| 1.4.1.1 | Search | Implementation of Search part of CA protocol |  |  |  |  |
| 1.4.1.2 | Channel | Implementation of Channel setup |  |  |  |  |
| 1.4.1.3 | Read | Implementation of PV Read |  |  |  |  |
| 1.4.1.4 | Analysis | Size, performance analysis to evaluate feasibility |  |  |  |  |
| 1.4.2 | Features | Implementation of CA Protocol Features. Features already developed in PoC are refined here. |  |  |  |  |
| 1.4.2.1 | Search | PV Search protocol |  |  |  |  |
| 1.4.2.2 | Channel | Channel setup |  |  |  |  |
| 1.4.2.3 | Read | PV Read command |  |  |  |  |
| 1.4.2.4 | Write | PV Write command |  |  |  |  |
| 1.4.2.5 | Event | Event notifications |  |  |  |  |
| 1.4.2.6 | Beacon | Server beacons |  |  |  |  |
| 1.4.3 | Release | Activities related to software release |  |  |  |  |
| 1.4.3.1 | Build | Compiling, Linking |  |  |  |  |
| 1.4.3.2 | Package | Packaging |  |  |  |  |
| 1.5 | Testing | All testing related activities |  |  |  |  |
| 1.5.1 | Test Benchmark | Development of test cases, and test program(s). |  |  |  |  |
| 1.5.2 | System Level Test | Performing system level tests |  |  |  |  |
| 1.6 | Documentation | Writing manuals |  |  |  |  |
| 1.6.1 | Installation Manual |  |  |  |  |  |
| 1.6.2 | User Manual |  |  |  |  |  |

# Project Strategy

The challenges are:

* The limited resources on the platform.
* The Channel Access Server Library (CASL) from EPICS cannot be used. The protocol will have to be implemented mostly from scratch.

The strategy is:

* Divide the work between a Soft IOC and the embedded controller, just like it is being done now. The embedded controller will provide read, write, and notifications for the PVs, and the soft IOC will take care of the rest.
* Knowledge-transfer on DEBROS. Understand DEBROS and develop documentation for posterity.
* Investigate if code from CASL or any other open-source implementation can be used as the starting point
* Develop a proof-of-concept. Evaluate the resource needs. Terminate if the solution is not feasible.

## Critical Success Factors

* Availability of DEBROS Subject Matter Expert (SME)
* Code optimization

## Project Life Cycle

The project will not be divided into subprojects. It will go through the activities as described on Quark. During planning or executing phases, if it becomes evident that the solution cannot be implemented, the project will be terminated.

## Product Life Cycle

* The first implementation will be a proof-of-concept (POC): Version 1.0 b0. If it is found that the implementation is not feasible, the project will be terminated.
* The lessons from the POC will be used to design the data structures and algorithms to optimize resource usage
* Test Plans will be developed to test and benchmark the solution for acceptance
* Functionality will be added and released in iterations (1.0b1 and 1.0)
* The NSCL development life cycle model will be followed as described in Quark. It will use an Agile approach at the beginning, and then move to a more predictive model with each iteration.

# Project Schedule

The project is estimated to take 943 person-hours.

## Schedule Management

# Project Budget

The project is estimated to cost the equivalent of 943 person-hours. There are no other material or procurement costs.

## Cost Management

# Human Resource Management

## Organization

* Sponsor: John Vincent
  + Customer: Kelly Davidson
  + Project Leader: John Priller/Vasu Vuppala
    - Project Coordinator: Vasu Vuppala
    - Design Team: Vasu Vuppala, John Priller
    - Development Team: Vasu Vuppala, John Priller
    - Documentation Team: Vasu Vuppala, John Priller
    - Subject Matter Expert – DEBROS: Mark Davis
  + Quality Manager: Vasu Vuppala
    - Test Team: Vasu Vuppala, John Priller, Mark Davis
  + Configuration/Change Control Board (CCB): John V, Kelly D, John P

## Roles and Responsibilities

Refer to Quark for the roles and responsibilities. Any additional roles are listed below.

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Subject Matter Expert  (SME) - DEBROS | Provide consultation and training in DEBROS |
|  |  |
|  |  |
|  |  |

### Responsibility Assignment Matrix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Role 1** | **Role 2** | **Role 3** | **Role 4** | **Role 5** | **Role n** |
| **Work Package 1** |  |  |  |  |  |  |
| **WP 2** |  |  |  |  |  |  |
| **WP3** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **WP n** |  |  |  |  |  |  |
|  | | | | | | |

## Resource Loading and Release Plan

# Quality Management

## Quality Metrics

On DEBROS-based RCM-3200 controller:

* Throughput:
  + 10 read transactions per second
  + 10 write transactions per second
* Response Time:
  + 20 ms (approximately) for a read operation
  + 20 ms (approximately) for a write operation

## Quality Management Plan

* Define metrics
* Develop benchmarks to test completeness and performance
* Develop software to test with benchmarks
* Determine quality audit schedule. Perform audits/reviews as per Quark processes

## Quality Improvement Plan

* Collect defect data at requirements, design, and implementation stages

# Risk Management

## Risk Breakdown Structure

### Risk Register

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Risk** | **Category** | **Root Cause** | **Probability** | **Impact** | **Response** |
| 1 | Inadequate Performance | Technical | Limited resources on RCM | Medium | High. Will not meet requirements | Optimize or terminate |
| 2 | DEBROS Defects | Technical | DEBROS Limitations | Low | High. Will increase cost/effort. | Modify DEBROS or terminate |
| 3 | Technical Know-how (DEBROS & CA) | Technical | No precedence (CA), lack of knowledge (DEBROS) | Medium | High | Training |
| 4 | Schedule slippage |  | Operational exigencies | Medium | High | Assign part of work to Mark, if possible. |
|  |  |  |  |  |  |  |

# Communications Management

## Stakeholder Register

* John Vincent
* Kelly Davidson
* John Priller
* Mark Davis
* Vasu Vuppala

## Communications Plan

* Meeting Schedule: Weekly
* Performance Reports: Weekly

# Procurement Management

No procurements, in terms of material or human resources, are expected for this project.

# Configuration Management

## Configuration Items

The configuration items are:

* The deliverables
* Project documentation: RFW, Preliminary Project Plan, Project Charter, Project Plan, Change Requests, Project Closure Report
* DEBROS version and documentation
* Build Information: Compiler Version and configuration, required libraries, build procedure

## Change Control

The change control process outlined in Quark will be followed. Refer to Quark for details.

## Change Classification (Major vs Minor)

* During development of Version 1.0 b1: Any change requiring more than 1 person-weeks of effort will be considered major.
* During development of Version 1.0: Any change requiring more than 0.5 person-weeks of effort will be considered major.
* After a total of 1 person-month effort for changes, all changes will be considered as major
* Effort estimations must be approved by CCB

# References

1. EPICS CA Protocol
2. DEBROS Manual
3. Request for Work

# Glossary