A logo for a university

AI-generated content may be incorrect.

SCARAB: Functional Specification

For submission to SETU

Author: Stuart Rossiter

Student Number: C00284845

Course: BSc (Hons.) Software Development

Supervisor: Joseph Kehoe

Contents

[1. Introduction 3](#_Toc210724209)

[1.1 Target Users 3](#_Toc210724210)

[1.2 Technologies 3](#_Toc210724211)

[2. Functional Requirements 4](#_Toc210724212)

[2.1 Core Requirements 4](#_Toc210724213)

[2.2 Non-Core Requirements 4](#_Toc210724214)

[3. Context Diagram and Use Cases 5](#_Toc210724215)

[3.1 Context Diagram 5](#_Toc210724216)

[3.2 Use Case Diagrams 5](#_Toc210724217)

[3.2.1 PC Program 5](#_Toc210724218)

[3.2.2 SCARAB Device Use Case Diagram 5](#_Toc210724219)

[3.3 Brief Use Cases 5](#_Toc210724220)

[3.3.1 Dump Save 5](#_Toc210724221)

[3.3.2 Restore Save 5](#_Toc210724222)

[3.3.3 Swap Cartridge Port 6](#_Toc210724223)

[3.3.4 Check Health 6](#_Toc210724224)

[3.3.5 Identify Corruption 6](#_Toc210724225)

[3.3.6 Verify Checksum 6](#_Toc210724226)

[3.3.7 Test Pins 6](#_Toc210724227)

[3.3.8 Test Save Retention 6](#_Toc210724228)

[4. Metrics 7](#_Toc210724229)

# Introduction

The SCARAB (Save and Cartridge Aid Requiring Adapter Boards) is a diagnostic tool designed to check the overall health of retro game cartridges, and manage their saves through a GUI based PC program. Designed for the purpose of retro game preservation, the SCARAB will use an Arduino MEGA 2560 to interface with the cartridges, through cartridge port modules. These modules will be able to be swapped in and out, allowing for a wide variety of cartridge types, and future expansion.

## Target Users

The SCARAB will be designed for retro gaming enthusiasts and preservationists. Retro gaming enthusiasts would want to ensure that their video games remain playable, and would be interested in backing up their save games in case something does go wrong. Preservationists keep all sorts of cartridges in working order, preserving the hardware with the software, so they would make use of the SCARAB’s diagnostic capabilities.

## Technologies

To serve as an interface between the cartridge port modules and the PC, a microcontroller will be required, in this case an Arduino. Due to the number of pins required for cartridge ports such as the Game Boy Advance, or the Nintendo 64, an Arduino MEGA 2560 would be ideal, due to the sheer number of digital I/O pins on the board. Alternate boards would be usable, but would require latches or multiplexing. For the GUI based PC program, Python would be sufficient, as it can interface with the Arduino directly over serial, and the Tkinter library provides GUI functionality. Various electronic components will also be required, listed below:

* 2x 8-bit transceivers, for 5v to 3.3v logic conversion.
* USB-C port. The Arduino has a max mA output, so a separate USB will be used to power the cartridges.
* Cartridge ports, to interface with the cartridges.

# Functional Requirements

The functional requirements for the SCARAB are separated into 2 groups: Core and Non-Core.

## 2.1 Core Requirements

* Users should be able to run diagnostic tests on inserted cartridges.
* Users should be able to dump the save data from their cartridges.
* Save Data should be able to be flashed back to the cartridges.
* Users should be able to view and manage their various save files from the GUI program.
* Cartridge ports should be able to be removed, and swapped out with other cartridge port modules.

## 2.2 Non-Core Requirements

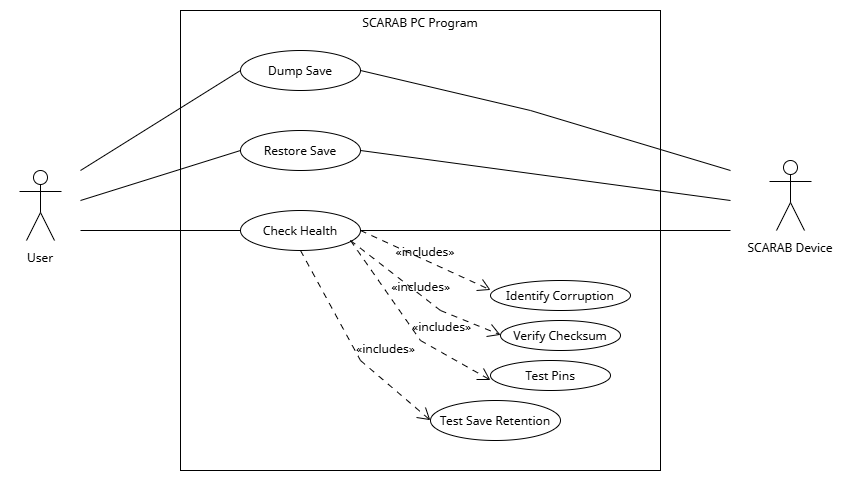
# Context Diagram and Use Cases

## Context Diagram

To be added.

## Use Case Diagrams

### 3.2.1 PC Program



### 3.2.2 SCARAB Device Use Case Diagram

A diagram of a device

AI-generated content may be incorrect.

## Brief Use Cases

### Dump Save

Actors: User, SCARAB Device

Description: This use case begins when the user wishes to dump the save file from the inserted cartridge. The user selects “Save Manager” from the sidebar. From there, the user selects the “Dump Save Data” option. The use case ends when the save data has been dumped from the cartridge.

### Restore Save

Actors: User, SCARAB Device

Description: This use case begins when the user wishes to restore a dumped save to the inserted cartridge. The user selects “Save Manager” from the sidebar. From there, they select the desired save, and press “Restore Save Data”. The use case ends when the save data has been restored to the cartridge.

### Check Health

Actors: User, SCARAB Device

Description: This use case begins when the user wishes to check the health of the inserted cartridge. The user selects “Health Check” from the sidebar. The user then selects the tests they wish to run, and clicks “Run Test(s)”. This use case ends when the results of the tests are returned to the user.

### Identify Corruption

Actors: User, SCARAB Device

Description: This use case begins when the user wishes to identify any corrupted data within a cartridge/memory card. The user selects “Health Check” from the sidebar. The user then selects “Identify Corrupted Blocks”, and clicks “Run Test(s)”. This use case ends when the result of the test is returned to the user.

### Verify Checksum

Actors: User, SCARAB Device

Description: This use case begins when the user wishes to verify the checksum of a cartridge ROM against a known dump. The user selects “Health Check” from the sidebar. The user then selects “Verify Checksum”, and clicks “Run Test(s)”. This use case ends when the result of the test is returned to the user.

### Test Pins

Actors: User, SCARAB Device

Description: This use case begins when the user wishes to verify the checksum of a cartridge ROM against a known dump. The user selects “Health Check” from the sidebar. The user then selects “Verify Checksum”, and clicks “Run Test(s)”. This use case ends when the result of the test is returned to the user.

### Test Save Retention

Actors: User, SCARAB Device

Description: This use case begins when the user wishes to test the save retention of the inserted cartridge. The user selects “Health Check” from the sidebar. The user then selects “Test Save Retention”, and clicks “Run Test(s)”. This use case ends when the result of the test is returned to the user.

### Determine Port

Actors: Cartridge Port

Description: This use case begins when a new cartridge port is inserted. The SCARAB probes the inserted board to determine what cartridge port the board holds. This use case ends when the Arduino has determined the port type.

### Set Pins

Actors: None

Description: This use case begins when the SCARAB determines a new port type. The SCARAB determines which pins to set as INPUT and OUTPUT. This use case ends when the pins are correctly set.

### Read Data

Actors: PC, Cartridge Port

Description: This use case begins when the SCARAB gets a read request from the PC. The SCARAB reads data from the cartridge. This use case ends when the requested data has been read.

### Write Data

Actors: PC, Cartridge Port

Description: This use case begins when the SCARAB gets a write request from the PC. The SCARAB receives the data and places it in a buffer. This use case ends when the requested data has been written to the cartridge.

### Return Data

Actors: PC

Description: This use case begins when the SCARAB needs to send data to the PC. This use case ends when the SCARAB has sent the data to the PC.

### 3.3.13 Release Port

Actors: PC, Cartridge Port

Description: This use case begins when the PC sends a request to the SCARAB to release the cartridge port. The SCARAB cuts power to the port and releases the security latches. This use case ends when the SCARAB sends a confirmation to the PC.

# Metrics