Maseeh College of Engineering and Computer Science

PORTLAND STATE UNIVERSITY

# PSU Capstone Project Proposal:

## Porting Existing Smart Ballot Box 2 Software to an Arduino Uno-R3 ATmega328P Platform


**Prepared By:**

Ali Saad

Jonathan Christian

Nick Long

Jiaqi Liu

**Prepared For:**

Joe Kiniry

Joey Dodds

Daniel Zimmerman



Revision Number: 5

Date of Revision: February 27, 2020

Date of Document Submission: February 27, 2020


FREE & FAIR

February 25, 2020

Dr. Joseph Kiniry, Dr. Daniel Zimmerman, and Dr. Joey Dodds
Free & Fair
421 SW 6th Ave #300
Portland, OR 97204

Dear Dr. Kiniry, Dr. Zimmerman, and Dr. Dodds,

## Overview / Executive Summary

Our sponsor, Free & Fair is a B Corp organization governed by a guiding mission statement, to make elections more verifiable. This leads them to create technological advances that help them achieve their mission. Over the last couple of years, they have been experimenting with a central component of the voting system - the smart ballot box. The result is a fully functioning Smart Ballot Box (SBB2019) prototype. It addresses serious concerns with electronic voting systems like fraud and manipulation, but it is expensive to reproduce as it features a high performance Xilinx FPGA, hindering replication. Our task is to create a more affordable version of their Smart Ballot Box (SBB), while maintaining its functionality, along with using widely available Commercial-Off-The-Shelf (COTS) components.

Voters have good reason to be concerned about how elections are conducted, particularly with the current trend towards using digital systems to automatically tabulate results. While elections could benefit greatly from this technological improvement, there remains serious issues to address. For example, the recent problems experienced by the Iowa Caucuses demonstrated technology selections are not up to the task of even collecting results, let alone providing a means to audit the outcome.

The main obstacle that Free & Fair is intending to solve with this capstone project is the high cost of the existing prototype that costs nearly $10,000. The current FPGA used as the CPU costs over $9,000 itself, making it an expensive system for voting centers. We have been given a task to find a more cost-effective CPU solution that will reduce the system cost below $500, but ideally around $300. Hence, we are proposing to use the Arduino Uno Rev3 based on the ATmega328P to interface with all the sensors and replicate the functionality of the SBB2019. This project will port over all the software, firmware and FreeRTOS to a UNO microcontroller and have a successful prototype built by May 15, 2020. A demonstration of our project will happen the first week in June to determine if this proposal and requirements meet Free & Fair's expectations.

# Product Design Specification

## Concept of Operations

The Smart Ballot Box is a key component of an electronic voting system. A user will receive a paper ballot that contains a special barcode. After they have selected their votes on paper, a voter would then approach our Smart Ballot Box with their completed ballot. They will place their ballot on the SBB and it will pull it in to verify its authenticity. Then, they will be prompted to select from one of two buttons. The first button is the Cast Button. If this button is pressed the ballot will be deposited in the ballot box. The second button is the Spoil Button. This button will scan the ballot and then return the ballot to the voter. In the current implementation of the SBB, the software only checks the ballot to see if it is legitimate or not. Some of the items contained in the barcode are which election it is for, the voters identity, and when the ballot expires. In future implementations the hope is that the SBB would also be able to record the actual votes of the voters when they are scanned. Another feature that we will not be exploring is networking ballot boxes together. They would be able to communicate the voting data to a central command and control computer.

### Stakeholders

- Free & Fair
    1) Joe Kiniry
    2) Daniel Zimmerman
    3) Joey Dodds
    4) Steven Osborn
- PSU Advisor
    1) Tom Schubert
- Capstone Team #4
    1) Ali Saad
    2) Jonathan Christian
    3) Nick Long
    4) Jiaqi Liu
- PSU Capstone Instructors
    1) Mark Faust
    2) Andrew Greenberg

### Intellectual Property (IP)

The IP and all work pertaining to the existing SBB2019 prototype is owned by Free & Fair, which they have shared with us for reference. As for the proposed modifications and additions created by the team, they will belong to both the team and Free & Fair. If our designs become implemented into the next iteration of SBB, they have agreed to recognize us for our contributions in return. We will be using Git to do signed commit, that way everyone can get credit for the work they do.

### Needs Identification

- Most voting centers cannot afford to spend $9000+ for a single ballot box

- The cost prevents private citizens from exploring the established security features and experimenting with how it works
- The current prototype uses some custom parts, not easily obtained outside of the manufacturer
- The current prototype can't be easily replicated and may not be safe to produce without training
- The current COTS components are not available everywhere and in limited supply (not long term)

**Objectives**

The main objective of this project is to create a more affordable version of the Smart Ballot Box created by Free & Fair by substituting an Arduino UNO in place of the Xilinx FPGA while keeping the original functionality and security features.

- Make an affordable smart ballot box
- Ensure the smart ballot box is easy to replicate
- Use the COTS components currently incorporated into the SBB2019
- Ensure the smart ballot box is safe to operate and manufacture
- Limit the number of custom parts and favor COTS components
- Promote experimentation of the smart ballot box by private citizens and explorers

**Requirements**

1. Functionality
    - The Product MUST have the same functionality as the original SBB2019 prototype
    - The Product MUST have two buttons to allow voters to cast or spoil
    - The Product MUST have an LCD screen
2. Performance
    - The Product MUST recognize a ballot within 20 seconds of being scanned
    - The Product MUST weigh less than 10 pounds
    - The Product SHOULD be smaller than 24"x 24"x 48"
    - The Product MUST house the ballots securely with a lock
3. Economic
    - The Product MUST cost less than $500
    - The Product SHOULD cost less than $300
    - The Product MAY cost less than $200
4. Power/Energy
    - The Product MUST be energy efficient
    - The Product MUST be able to supply 12V

5. Health and Safety
    - The Product MUST not use harmful or toxic materials
    - The Product SHOULD be of durable construction
    - The Product MUST not have sharp or jagged edges
    - The Product MUST be enclosed, no point of contact with electrical components
6. Environmental
    - The Product SHOULD be recyclable
7. Legal
    - The Product MUST abide by the law and not violate citizens rights
8. Political
    - The Product MUST remain neutral in all elections
9. Usability
    - The Product MUST not require training to operate
10. Documentation
    - The Product MUST have clear and detailed documentation
11. Marketing
    - The Product MUST be cost-effective
12. Reliability & Availability
    - The Product MUST execute cast or spoil correctly
13. Physical Operation
    - The Product MUST be usable by polling center employees that have no prior technical knowledge and little training.
14. Manufacturing
    - The Product MUST allow others to replicate the work
15. Maintainability
    - The Product MUST be easily serviced
16. Software Security
    - The time a ballot is submitted MUST be tracked
    - MUST recognize whether a ballot has already been scanned
    - Making sure that every QR code is different/randomly generated but yet has the contents it needs to like the time and what election it is
    - Count how many ballots are cast, that way if tampering occurs we know can verify how many ballots are in the SBB
    - A user MUST not be able to configure any of the peripheral devices, allowing access to the system

**Specifications**

Provided by Free & Fair for SBB2019

- The code from SBB 2019
- The bill of material from SBB 2019
- The components required
- The Smart Ballot Box 2019
- Mentorship

Exterior Hardware:

- Interactive dual button interface for testing and simulation of cast or spoil the ballot, as well as the motors and LCD screen. (CASCADIO Board)
- Arduino Uno

Software, an adaptation of already written code:

- FreeRTOS
- SBB Software
- Firmware
- I/O Mapped correctly

We will be reusing the software provided by Free and Fair. We will focus on modifying it to function properly with the ATmega328P chip aboard the Arduino Uno. We will have to edit the RTOS and SBB code provided to properly interface with the sensor blocks. These modifications should be made to all compatibility with our chip but contain the same necessary functionality of the original SBB2019.

**Deliverables**

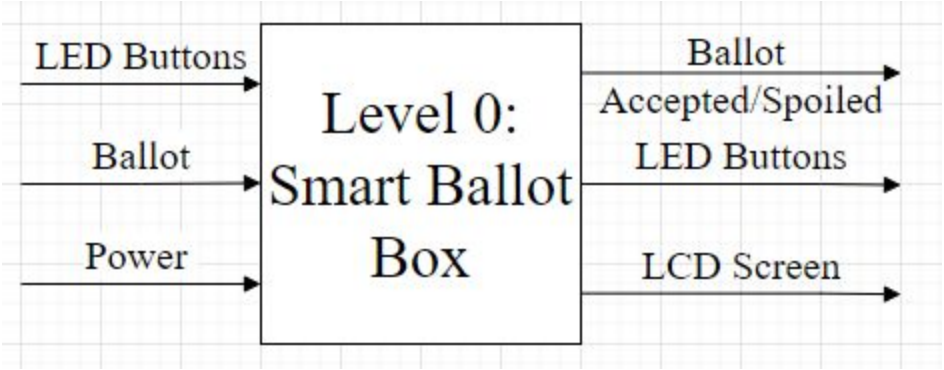
1. Our Smart Ballot Box 2 will have the same functionality as the original SBB2019 (except network logging and DEFCON debugging) as described in the Github BVS2019 project with the deployment version of our choosing.
2. The Arduino Uno will interface with the peripherals identically like the FPGA.
3. The system will be easy to replicate and build via the use of widely available COTS parts, with minimal modification required.
4. The cost will be below $500 to allow others to replicate our work
5. The overall design will allow for easy exploration, debugging, modification and experimentation for the future development of the Smart Ballot Box concept. This System is not a final product but apart from a more widespread effort to improve voting system technology.

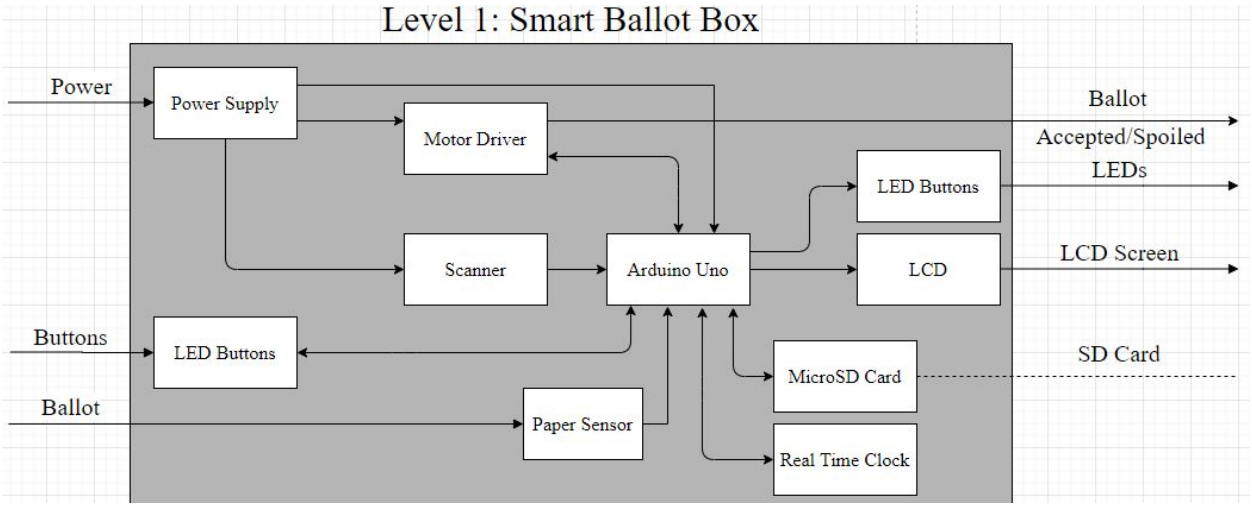
6. Project Proposal, Biweekly/Weekly Progress Reports (w/ milestones), a Final Report, Detailed Documentation, and a Capstone Poster.

These deliverables will be achieved by checking in with our advisor and sponsor regularly while updating them on the progress we are making.

**Initial product designs**

As mentioned above, we are proposing to make a more affordable version of Smart Ballot Box. The way we are going to achieve it is to analyze and learn from the Github provided by Free & Fair and try to make a SBB driven by an Arduino Uno.




**Verification Plans**

Level 1: Unit Testing

Arduino Uno communicates with SBB
Cast or Spoil Button Performance Testing
Motors Performance Testing
LCD Screen Performance Testing

Paper Sensors Performance Testing
Power Supply Performance Testing
Real Time Clock Performance Testing
MicroSD Card Performance Testing

Level 2: Integration Test

Test interactions between each unit

Level 3: System Testing

Ballot Scan Time Testing
Cast or Spoil Performance Testing

## Product Management Plan

### Timeline, with milestones

Please refer to the attached Project Schedule for milestones and general timeline. https://docs.google.com/spreadsheets/d/1V53Ma0pD8M68XWJ6WGMeonGLqihcLyoa/edit#gid=1525204150

### Budget and Resources

Our overall implementation of the Smart Ballot Box will cost around $50. For a more specific break down of the cost please refer to the Bill of Materials attached below. We will source most of the smaller electronic components from the EPL and Digikey. While we will use Amazon for the larger items. Before purchasing any new component we will first check with Free & Fair to see if they have a surplus of needed items.We also have some constraints on access to the SBB2019 as a resource.

Free & Fair will not allow us access to their original SBB hardware outside their office due to the price and weight. We will have access to work space during regular business hours once we get access to the building allowing us to enter without having to sign in as guests. This means we can only work between 9:00 am to 5:00 pm Monday through Friday. Outside of these times, we need to work in a predetermined development environment on our personal computers.

### Team and development process

Jonathan and Jiaqi are the Electrical Engineers in the group so they will be focusing their skills on the hardware. Nick and Ali are Computer Engineers so they will be focusing their skills on the software components of the project. When it is time to communicate to the industry and faculty advisor it will be done by Ali, our team leader. For the project we will be using GitHub and Keybase to collaborate on work.

Sincerely,

Ali Saad, Jonathan Christian, Jiaqi Liu, Nick Long

Included in the SBB as stated in the Bill Of Materials:

| Qty | Value | Device | Package |
|---|---|---|---|
| 1 | YELLOW | LEDCHIP-LED0603 | CHIP-LED0603 |
| 1 | RED | LEDCHIP-LED0603 | CHIP-LED0603 |
| 2 | BLUE | LEDCHIP-LED0603 | CHIP-LED0603 |
| 3 | GREEN | LEDCHIP-LED0603 | CHIP-LED0603 |
| 2 | .1" | PINHD-2X6 | 2X06 Standard 2.54mm pitch header |
| 1 | .1" | PINHD-1x16 | 1x16 Female Pin header 2.54mm |
| 2 | Adjustable | V_REG_LM1117SOT223 | SOT223 |
| 2 | .1uf | C-USC0603K | C0603K |
| 1 | .1uf/50V | C-USC0805K | C0805K |
| 1 | 1.5k | R-US_R0603 | R0603 |
| 20 | 10k | R-US_R0603 | R0603 |
| 2 | 10pf | C-USC0603K | C0603K |
| 2 | 10uf | C-USC1206K | C1206K |
| 2 | 1uf | C-USC0603K | C0603K |
| 2 | 2.2k | R-US_R0603 | R0603 |
| 2 | 220R | R-US_R0603 | R0603 |
| 1 | 22uF / 63V | CPOL-USD | PANASONIC_D |
| 1 | 33k | R-US_R0603 | R0603 |
| 7 | 470R | R-US_R0603 | R0603 |
| 2 | 4.7k | R-US_R0603 | R0603 |
| 1 | 32.768khz | ABS07W-32.768KHZ-D-1-T | ABS07AIG32768KHZ71T |
| 1 | 10MM_SM_COIN_CELL_CLIP | 10MM_SM_COIN_CELL_CLIP | 10MM_SM_COIN_CELL_CLIP |
| 1 | DRV8871DDA | DRV8871DDA | SOIC127P600X170-9N |
| 1 | DS1338-33 | IC_RTC_DS1307 | SO-8 |
| 1 | LM393D | LM393D | SO08 |
| 1 | MICROSD | MICROSD | MICROSD |
| 4 | PMV65UNEAR | NMOSSOT23 | SOT-23 |
| 1 | POWER_JACKSLT | POWER_JACKSLT | POWER_JACK_SLOT |
| 2 | SPDT_TOGGLE_BUTTONMICRO | SPDT_TOGGLE_BUTTONMICRO | BTN-PUSH-PUSH-PS-2214-L |
| 1 | ARDUINO-NOHOLE | ARDUINO-NOHOLE | ARDUINO-NOHOLE |
| 1 | OLED | 1X4-CLEANBIG | 1X04-CLEANBIG |
| 1 | RED | MOMENTARY-SWITCH-SPST-LED-PTH-12MM | TACTILE_SWITCH_LED_PTH_12MM |
| 1 | GREEN | MOMENTARY-SWITCH-SPST-LED-PTH-12MM | TACTILE_SWITCH_LED_PTH_12MM |
| 2 | | CONN_02POLAR_LOCK | JST-XH 2in 2.54mm pitch |
| 1 | | CONN_03POLAR_LOCK | JST-XH 3in 2.54mm pitch |

| Parts | Description | MPN |
|---|---|---|
| LED1 | LED | |
| LED2 | LED | |
| LED5, LED6 | LED | |
| LED3, LED4, LED7 | LED | |
| PMOD0, PMOD1 | 2x6 Pin header, through hole | |
| U$1 | 1x16 Female Pin header 2.54mm | |
| U2, U3 | Voltage Regulator LM1117 | |
| C9, C12, C1, C2, C13 | CAPACITOR, American symbol | |
| C7 | CAPACITOR, American symbol | |
| R30 | RESISTOR, American symbol | |
| R2, R4, R5, R6, R7, R8, R9, R13, R15, R16, R17, R19, R20, R21, R22, R23, R27, R28 | RESISTOR, American symbol | |
| C4, C5 | CAPACITOR, American symbol | |
| C8, C11 | CAPACITOR, American symbol | |
| C3, C10 | CAPACITOR, American symbol | |
| R1, R3 | RESISTOR, American symbol | |
| R11, R14 | RESISTOR, American symbol | |
| C6 | POLARIZED CAPACITOR, American symbol | |
| R18 | RESISTOR, American symbol | |
| R12, R10, R26, R29, R31, R32, R35, R36, R37, R38 | RESISTOR, American symbol | |
| R24, R25, R34, R35 | RESISTOR, American symbol | |
| Y1 | Abracon 32.768kHz Crystal 2-SMD, No Lead 3.2 x1.5 x 0.9mm | 815-ABS07W32.768KD1T |
| G1 | Battery cell clip for CR927 | 3030 |
| IC4 | 3.6A Brushed DC Motor Driver | 595-DRV8871DDA |
| IC1 | Real Time Clock | DS1338-33 |
| IC2 | Dual Comparator | LM393DR |
| U1 | Micro-SD / Transflash card holder with SPI pinout | 2908-05WB-M |
| T3, T4, T5, T6 | MOS FET | PMV65UNEAR |
| J2 | Power Jack Connector 2.1x5.5mm | PJ-202A |
| S3, S4 | Toggle button push/push | PS-2214-L NS PA |
| U$2 | Arduino Diecimila/Duemilanove | |
| OLED | 4-pin connector | |
| S1 | Momentary Switch (Pushbutton) - SPST - w/ LED | |
| S2 | Momentary Switch (Pushbutton) - SPST - w/ LED | |
| MOTOR, PWR | | |
| SENSE1, SENSE0 | | |