Communication in Physical Sciences, 2026, 13(2): 246-266

Design and Implementation of a Cost-Effective Electronic Voting Machine Using Arduino Microcontroller

Emmanuel Acquah

Received: 3 November 2025/Accepted: 10 February 2026 /Published: 20 February 2026
https://dx.doi.org/10.4314/cps.v13i2.5

Abstract: *Electronic voting systems are a form of technological advancement in the democratic process, but in developing countries, their use is limited by their prohibitive prices and safety issues. This paper proposes the design, fabrication and testing of a low-cost Electronic Voting Machine (EVM) based on the Arduino Uno microcontroller technology in electoral situations of resource scarcity. The study deals with inherent inefficiencies of the traditional paper-based voting, such as long processing time, vulnerability to electoral fraud, high expenses, and long compilation of voting results. Features incorporated in the prototype include a 16×2 LCD, eight push buttons to select a candidate, authenticate with a password and control power, 5V DC 5V DC. Circuit simulation was performed with Proteus Professional 8.2 and board development was performed with Arduino IDE with C/C++ programming. Testing showed immediate registration of votes, one-second of tallying, less than three minutes of voter processing and another absolute over-voting protection with software-enforced single-access control. Cost analysis showed total expenditure of GH140.00 (US$31.60) total hardware expense which is a considerable affordability as compared to commercial systems of EVM. The standalone structure also reduces the vulnerability of cybersecurity by removing the network connectivity. The results provide technical and economic viability of Arduino-based electronic voting towards electoral modernization in developing economies, which assist to boost democratic processes through better efficiency, transparency and fiscal sustainability.*

Keywords*: Electronic voting machine; Arduino microcontroller; Electoral systems; Cost-effective design; Vote automation; Electoral transparency*

Emmanuel Acquah
Electronics Engineering Department, Norfolk State University, Norfolk, Virginia, United States.
Email:
e.acquah122756@spartans.nsu.edu
https://orcid.org/0009-0005-0848-5924

1.0 Introduction

Integrity in the electoral processes is a pillar in democratic governance, but voting systems have been evolving gradually in most parts of the world. Conventional paper-based ballot systems are known to create a lot of operational pressure on the electoral management institutions and the citizens. Voting in Ghana and most developing countries involves the use of traditional voting methods where voters are required to physically stamp their fingerprints on paper ballots, place the ballots into closed ballot boxes and wait hours or days to have their votes tabulated manually- a time-consuming process based on the volume of votes (Alvarez & Hall, 2008). The 2016 general elections of Ghana needed about GH fragment 826 million (US186 million), whereby a significant amount of it was allocated to the purchase of ballots, printing, transportation, polling stations, and staff compensation (Electoral Commission of Ghana, 2016).

In addition to the financial aspects, the conventional systems have operational issues that undermine the quality of the elections. The voter queuing often lasts a number of hours, which in effect disenfranchises those citizens who cannot afford to spend a significant amount of time without a job (Edelstein & Edelstein, 2010). Counting by hand opens up possibilities of unintentional mistakes and intentional

manipulation, and the controversial counts may sometimes lead to electoral violence or even years of legal wrangling that destroy democratic trust (Norris, 2014). Voting that is intensive on paper has environmental costs based on deforestation and the problem of safe disposal of the used materials, as well as keeping secrecy (Jones, 2009).

Towards the end of the twentieth century, electronic voting technologies came in with the promise of meeting these limitations by offering digital vote capture, automated tabulation and improved auditability (Mercuri, 2002). In Brazil, India, and some U.S. precincts, Direct-Recording Electronic (DRE) showed a possibility of a shorter time to count, reduced spoiled ballots, and enhanced accessibility (Brunazo Filho *et al*., 2015). Nevertheless, electronic voting was met with a heated discussion about the lack of cybersecurity, the lack of voter-verifiable paper trails, the opaqueness of the proprietary software, and the concentration of the electoral infrastructure in the hands of a limited number of commercial vendors (Schneier, 2004; Wolochok *et al*., 2012). The perceived or proven weaknesses of the Netherlands against the United States increased the doubts on whether electronic systems actually increase the integrity of elections or avenues of fraud (Gonggrijp & Hengeveld, 2007).

To developing countries that consider the electoral modernization, these arguments have some extra cost-effectiveness and technological dependency level. Electronic voting systems used in commercial applications generally cost thousands or tens of thousands of dollars apiece, and also include maintenance contracts, software licenses, and specialized technical support-something that an electoral commission in extreme financial constraint situations cannot afford (Herrnson *et al*., 2008). The use of proprietary systems poses issues regarding technological sovereignty, ability to audit independently, and lock-in of the vendor that means the inability to be flexible in the future (Gibson *et al*., 2016). These factors led to the study of open-source and low-cost solutions based on the extensive use of microcontroller platforms that could be used to provide core electoral capabilities at lower costs (Verma, 2012; Hussain *et al*., 2016).

In 2005, the first platform was the open-source, electronics prototyping based on Atmel AVR microcontrollers, Arduino platform, which had the potential to become an easily available EVM platform (Banzi & Shiloh, 2014). The design of Arduino makes it easy to use with simplified programming interfaces, voluminous community documentation and modular hardware that can be used with different integrations. The low cost of its boards, typically around US$20-30, and the open-source license that allows unlimited modification of its software fits resource-wise and in terms of sovereignty that is specific to developing nation applications (D'Ausilio, 2012). In the past, studies had shown the feasibility of Arduino in embedded systems applications such as in environmental monitoring, industrial automation and teaching instrumentation (Teikari *et al*., 2012).

However, the prototypes of the Arduino-based voting systems had an unequal balance in terms of security, functionality, cost, and usability. The first work by Verma (2012) created an EVM that uses Arduino but with microcontrollers with no EEPROM, which can only hold most votes during interruptions in power, which is an important auditability standard. Hussain *et al*. (2016) suggested RFID-based smart EVM and biometric authentication and GSM transmission, albeit by adding luxurious components, whose costs were much higher than the availability of the resources to deploy it. Murali *et al*. (2016) designed EVM with the integration of fingerprint biometrics and microcontroller processing, which possesses improved security at the cost of affordability because biometric sensors are relatively expensive.

The study solves optimization problems by designing and developing a minimalist, but

functionally complete, Arduino-based EVM that is developed to be affordable, but meets the basic electoral needs, including accuracy, auditability, and security. The architecture is specifically designed to focus on the broadly available parts of developing nation electronics markets, avoids reliance on proprietary software, and focuses on operational simplicity with reduced training needs. In place of integrating all of the conceivable security features, the design is aimed at reliably performing the essence of a basic voting system: secure vote capture, accurate tallying, tamper resistance and instant compilation of results at target hardware costs less than US$50 per unit.

The design of an EVM architecture with commonly available Arduino parts, demonstrating functional parity with traditional systems in terms of vote accuracy and auditability, in addition to being faster and more efficient, is the threefold research objective. Second, design, simulate and test this design with virtual simulation and physical prototyping with careful testing of performance on factors such as vote processing time, counting accuracy, security against common attacks and reliability with realistic conditions. Third, do a thorough cost-benefit analysis of the developed system versus paper-based voting and commercial electronic options, considering economic feasibility in the context of the usual fiscal limitations of a developing nation.

The importance of this work is not just about the direct contribution to the current technical issue in the form of a practical low-cost voting prototype. The study helps to expand the scope of technological sovereignty discourse and proper selection of technology in developing countries by showing that it is possible to implement critical electronic voting features with open-source software and off-the-shelf hardware. Electronically documented design decisions, component selection and implementation present replicable templates to Indigenous voting technologies, when electoral commissions or institutions require localized indigenous voting technologies. In addition, cost-benefit analysis provides facts that may aid policy discussion on electoral modernization and so may affect resource allocation in which each dollar will compete against essential developmental priorities.

The article is structured in the following way: Section 2 provides a theoretical framework that reviews the typologies of electronic voting, design requirements that rule efficient electoral technology, and microcontroller platform functionality associated with voting systems. Section 3 describes system design architecture, choice of components to use, power supply, software development and testing of the system simulated and prototyped. Section 4 introduces and explains the findings of the experiment, comparing the measurements of performance, the cost analysis, security features, and the relative location with respect to previous studies. Section 5 is a conclusion, limitation and future direction of investigation.

1.1 Theoretical Framework

1.1.1 Electronic Voting Systems Classification and Evolution

Electronic voting has a variety of technological methods that are distinguished by vote capture methods, recording media, user interfaces and tabulation methods. Knowledge of this taxonomy would place the existing research in a larger context of electoral technology and would identify the various design tradeoffs of different architectures. The development of voting technology is a kind of constant tradeoff between competing values: ballot secrecy or verifiability, efficiency of automation or transparency, cost reduction or security strength (Mercuri, 2002).

The first attempts at automation were punch-card voting systems that began to appear in the United States in the 1960s. The voters scored the cards with styluses, and the pattern of holes thus made was read by counting machines, either optical or mechanical. Although the punch-card

systems minimized the number of people required to count by hand, they became susceptible to incomplete perforations (the notorious hanging chad that affected the 2000 U.S. presidential elections), cumulative mechanical damage that impacts the results of reading, as well as intentional tampering with the results by pre-scoring or punching twice (Saltman, 2006).

The optical scan voting system is a system of marking paper ballots and is analyzed with the help of image recognition technology. Voting is conducted by voters either by means of standardized ovals or checkboxes on a machine-readable form, which is fed to optical scanners that identify the position of marks and cast the vote in a digital format (Jones & Simons, 2012). Optical scan has its merits, such as paper ballot is saved in case of manual recounting, it is mostly resistant to software manipulation, and voters are accustomed to the paper-based interface. Nevertheless, systems are still vulnerable to issues of ballot design error that bewilders voters, scanner mal-alignment and the fact that they still need to pay people to continue printing ballots (Herrnson *et al*., 2008).

Direct-Recording Electronic (DRE) voting machines were commercially introduced in the 1990s, and the ballot options are shown on electronic displays, often touchscreens or button-activated screens, and recorded directly on digital memory, and do not generate paper intermediaries as votes are cast (Brunazo Filho *et al*., 2015). The benefits of DRE are multilingual interfaces, audio prompting as an accessibility feature to visually impaired voters, removal of ambiguous marks that can bedevil the paper ballots, and real-time result tabulation during the closing of the poll (Kohno *et al*., 2004). However, DRE systems have spawned huge controversy over being vulnerable to software attacks, lack of voter verifiable paper trails in a vast majority of systems, and the inability to perform viable post election audits in instances where the votes are recorded digitally (Feldman *et al*., 2007).

The Voter-Verified Paper Audit Trail (VVPAT) systems were developed as a reaction to the shortcomings of DREs, and tried to find the balance between electronic efficiency and paper-based auditability (Mercuri, 2002). Machines with VVPAT print paper records of each vote, which are checked by voters through windows before being cast and paper records are retained in case of a manual recount. Although VVPAT partially satisfies the needs of certain pure DRE issues, it adds more complexity and requires more expensive hardware since it uses printer mechanisms and casts uncertainty on the actual verification of paper records by voters (Goggin *et al*., 2012).

Internet voting allows rounding votes by means of web browsers or specific programs on either open or closed networks (Olembo *et al*., 2013). In 2005, the first nationwide implementation in Estonia showed that it was technically possible to cast ballots remotely electronically (Vassil *et al*., 2016). Internet voting, however, faces harsh security risks such as client malware that can potentially change votes before encryption, distributed denial-of-service attacks that hinder the voting time, and basic challenges in ensuring both voter anonymity and preventing vote selling or coercion in insecure settings (Springall *et al*., 2014).

There are various insights that can be shed light on through this review in relation to the present research. To start with, voting technology does not exist that is able to achieve all the desirable qualities, designers have to compromise on conflicting priorities. Second, even the most advanced technological solutions do not always imply the most effective solutions in all situations, and less developed systems can be more compatible with resources and expectations of voters. Third, there are still paper-based audit mechanisms that demand considerable legitimacy, implying that pure electronic systems need exceptionally high levels of transparency and verifiability to gain acceptance by the populace. The EVM built

on the Arduino board takes the same role as the DRE systems but is intentionally simplified to reduce the cost, and retains the essential features required in a controlled polling station setting.

1.2 Design Criteria and Requirements for Electoral Technologies

The requirements of a good voting system should meet rigorous constellations of occasionally conflicting requirements in the areas of technical performance, security properties, usability properties and operational constraints. Academic literature gradually established and improved such criteria, going beyond the crude counting speed productions to include complex ideas about auditability, accessibility and voter confidence (Jones & Simons, 2012; National Institute of Standards and Technology, 2007).

The most important domain of requirement is security. Voting technologies should maintain ballot secrecy that guarantees that no mechanism exists between individual voters and vote options, where vote buying, coercion or retaliation is prevented (Chaum, 2004). At the same time, the system should ensure that it authenticates voters, removing the possibility of unauthorized voting and remain anonymous at the same time. Combinations of authentication and casting of ballots are often not possible without separating the two. Protections of data integrity should prevent tampering with, deleting and adding entries to the recorded votes by a malicious individual with physical or electronic access, so that it needs cryptographic key systems and physical security measures (Karlof *et al*., 2005).

Accuracy requirements indicate that the systems must be capable of accurately recording the intent of the voter and properly tabulating the results without any systematic error that would distort the results of the election. To attain high accuracy would require proper user interface development, error correction and detection systems as well as quality control in the production and configuration (Everett *et al*., 2008). Voter interaction studies of various ballot designs showed that design choices that appear insignificant to many, such as the location of buttons, fonts, color coding, confirmation messages, etc., have a significant effect on error rates, especially among older or less educated voters (Herrnson *et al*., 2008).

The concept of usability involves a wide range of considerations such as the ease of learning by first-time voters, efficiency by experienced voters, memorability in multi-year elections, error prevention and recovery, and subjective satisfaction (Goggin *et al*., 2012). The principles of universal design encourage the concept of systems that are accommodative to various abilities such as the visually impaired, motor-limited, mentally impaired, and language diverse (Runyan, 2007). Multiple modalities of interaction usually have to be used to achieve wide accessibility, which complicates and makes the implementation expensive.

Auditability: the ability to look-glass check that votes have been properly documented and tabulated was seen as a key issue, especially in fully electronic systems which have no inherent paper trails (Benaloh, 2006). Audit mechanisms are between simple redundancy and complex cryptography methods that allow statistical verification without showing any votes. Threat model is a critical element of audit effectiveness: audits that are intended to detect random hardware failures might not work against highly engineered attacks that target results in a selective manner and that have plausible consistency (Goggin *et al*., 2012).

Cost-effectiveness analysis should not just focus on the initial capital expenditure but should take into consideration the total ownership cost, such as maintenance, consumables, training and eventual replacement (Herrnson *et al*., 2008). In the context of developing nations, the relevant comparison is not between various electronic technologies but between electronic and traditional paper-based systems, which

introduce high costs that are usually not attributed properly, at the cost of printing, distribution logistics, secure storage, and counting of the results when done by hand (Caarls, 2010).

These multi-dimensional needs are the ones guiding the design philosophy behind this study. The developed system does not aim to optimize on all performance criteria at the same time, usually making resource-constrained settings prohibitively expensive: instead, core requirements that matter more than others in the basic electoral integrity are met more reliably: vote accuracy, counting reliability, over-voting resistance, auditability with persistence and cost-effectiveness with component reduction. Additional security system features such as biometric authentication, cryptography vote protection, or result transmission over the network are not included because they are intentionally not included to keep the system simple and affordable.

1.3 Microcontroller Technology and Arduino Platform Capabilities

Microcontrollers are programmable specialized integrated circuits that comprise of programmable processing cores, memory subsystems and customizable input/output peripherals within single semiconductor packages tailored to embedded control applications (Wilmshurst, 2009). In contrast to the general-purpose microprocessors, which aim to be able to achieve the highest computational performance, microcontrollers are oriented on low cost, low power use, real-time response, and on integration of typical peripheral capabilities that do not demand large amounts of external circuitry (Valdés-Pérez and Pallas-Areny, 2009).

The Arduino platform was brought to market in 2005 by Massimo Banzi and coworkers at Interaction Design Institute Ivrea and democratized microcontroller-based prototyping by making specific design decisions that focused on non-expert accessibility (Banzi & Shiloh, 2014). Arduino boards combine Atmel AVR microcontrollers with voltage regulators, USB programmers, and unified pin headers that allow connecting the board with sensors and actuators without the need to solder the connection. Arduino IDE is a complementary software offering simplified C/C++ language programming environments that hide the low-level register manipulation of hardware behind simple function calls, a comprehensive collection of libraries, and simple upload programs that have been boiled down to just plugging into a USB cable (D'Ausilio, 2012).

Arduino Uno, based on the ATmega328P microcontroller, is the most commonly deployed variant of Arduino. The ATmega328P has 32 KB flash memory, 2 KB SRAM, and 1 KB EEPROM memory storage, sufficient to store programs, runtime data, and persistent data respectively, across power cycles, which in comparison is very modest by modern-day standards but more than sufficient to store the program, data, and other information required by most embedded control systems (Atmel Corporation, 2015). The microcontroller has a clock frequency of 16 MHz, which offers throughput adequate to process sensor data in real-time, manage user interface and handle communication protocols without consuming unnecessary power.

The ATmega328P has communication peripherals such as UART, used in serial communication, and SPI, used in high-speed communication with other specialized integrated circuits, among others, and I 2 C bus, which is used in multi- device networks (Atmel Corporation, 2015). These communication methods allow Arduino Uno to connect with various peripherals such as simple sensors or complex ones, such as GPS receivers or wireless transceivers. The ability to mix a wide range of different types of peripherals on a single application allows Arduino to be of special use in prototyping of complex systems such as voting machines, which combine user input, visual feedback, data storage, and result reporting.

The open-source licensing of the Arduino ecosystem under the Creative Commons and GNU General Public License ensures that users can study, modify and distribute both the hardware design and software without any royalty fees or vendor lock-in (Gibb, 2014). This openness to intellectual property is especially important in issues concerning electoral applications because transparency and the opportunity to be scrutinized by the people were important aspects of trust. Complete system schematics can also be reviewed by electoral commissions or an independent audit, and the code used to implement the vote counting logic can also be reviewed, and it should not contain any hidden functionality, something that is frequently virtually impossible with commercial proprietary voting systems.

Past studies have shown that Arduino is technically viable in various embedded systems applications for voting machine needs. The long-term stability and accuracy of the sensor data acquisition by Arduino were tested in the field of environmental monitoring (Teikari *et al*., 2012), whereas the real-time response suitability to control the process was tested in the industrial automation sector. Educational instrumentation projects proved the applicability of Arduino to powering alphanumeric displays, processing button inputs, and controlling data storage, of which peripheral interactions are the key to a voting machine working (D'Ausilio, 2012).

Basic Arduino has security drawbacks such as no hardware cryptographic accelerators that can be used efficiently to encrypt votes, no facility to provide secure booting, and vulnerability to read-out attacks by any opponent that can access the physical hardware (Chattopadhyay *et al*., 2017). These restrictions limit proper deployment situations to physical security polling stations as opposed to unsecured internet voting. Existing studies specifically adopt this limitation, and their design is specifically based on the deployment of polling stations, in which the presiding officer is constantly observed, and physical security measures are implemented to avoid unauthorized access to hardware.

2.0 Methodology

2.1 System Architecture and Design Rationale

The EVM architecture actively embraces a minimalism design philosophy with the focus on functional adequacy, component availability, and cost-effectiveness without compromising the necessary demands of electorally acceptable and secure as well as auditable solutions. The system block diagram in Fig. 1 shows key components and connections between them. Architecture The architecture consists of five main subsystems: (1) Arduino Uno microcontroller to act as central processing unit, (2) 16 x 2 character LCD to provide visual feedback, (3) eight tactile push buttons to allow the selection of the candidates, (4) regulated 5V DC power supply, and (5) perforated prototyping board to provide mechanical structure

Arduino Uno has been placed in the middle as shown in Fig. 1, of all the information flow between the input (push buttons) and output (LCD, serial monitor) devices, storage (internal EEPROM) and power distribution. This centralized architecture reduces system logic complexity by reducing the count of vote validation; storing functions to a single programmable element, and eliminating coordination complexity found with the distributed architecture that uses multiple processing nodes.

The choice of Arduino Uno on other microcontroller platforms represents a number of considerations. First, Arduino Uno is the most popular and published Arduino type in the world, and it can be purchased with high precision by official distributors and third-party providers at very low costs (Banzi & Shiloh, 2014). This availability is important when the contexts of developing nations are to be considered, where supply chain constraints can restrict access to specialized components. Second,

ATmega328P offers fully sufficient processing capability; vote counting consists of a few simple integer arithmetic operations that can be comfortably run on an 8-bit CPU, whereas interface administration only needs a small amount of processing bandwidth (discrete button presses and character-generated displays).

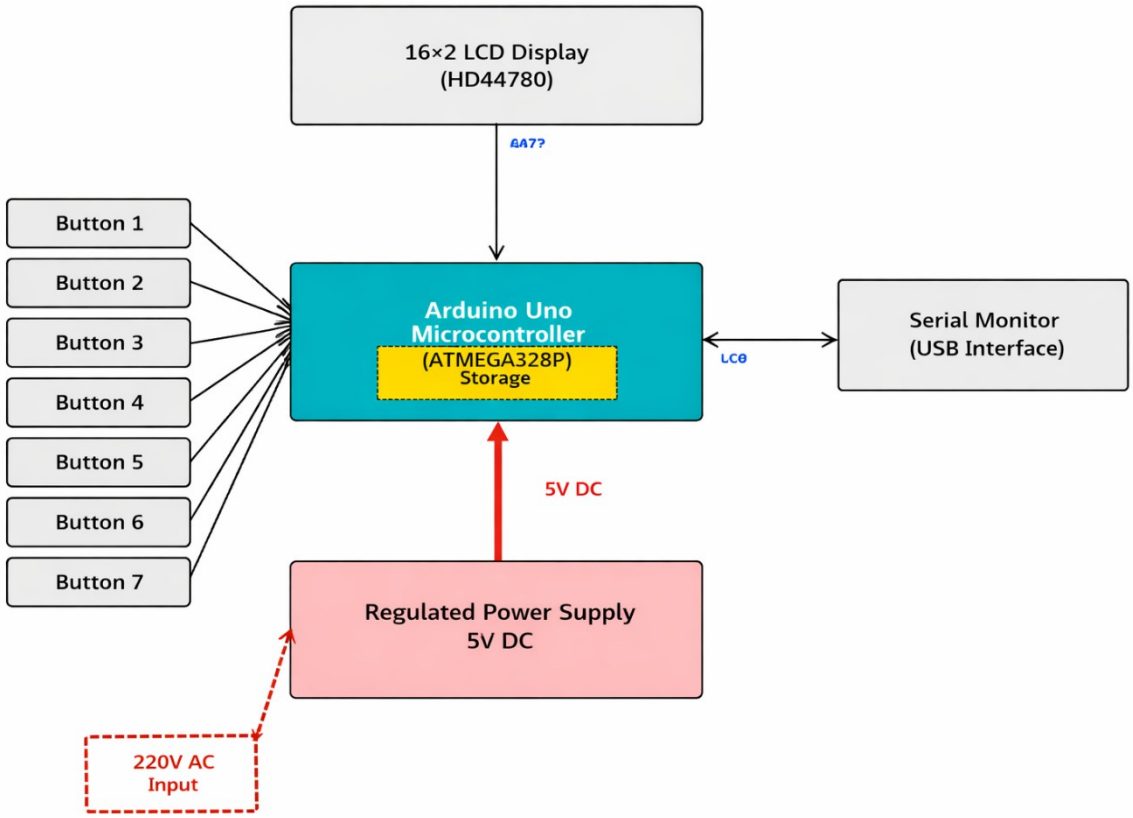

Fig. 1: System block diagram of Arduino Uno microcontroller connected to LCD display, eight selection buttons of candidates, and a regulated power supply. The Arduino reads button presses, controls LCD display, and stores votes in EEPROM, and transmits findings over the serial interface.

Third, Arduino Uno has a vast community base that guarantees availability of code samples, troubleshooting resources and design guidance to hasten the development speed and transfer of knowledge to local technicians who service deployed systems.

The 16 2 LCD option is a trade-off between legibility, price and ease of interfacing. These screens offer enough room on the display to show system status messages, voting instructions, candidate selections, and tallies of results without the need to have sophisticated graphical displays (Wilmshurst, 2009). The HD44780 industry standard controller chip has well-documented command protocols and wide support of Arduino libraries, allowing the control of the displays to be done with simple function calls instead of low-level bit manipulation (Banzi & Shiloh, 2014).

The voter interface mechanism is a feature of push buttons and there is one button dedicated to a candidate. This prototype of the research uses eight buttons, which support the use of contests involving up to eight candidates – the setup is appropriate in many elections to single offices. Instead of capacitive touch sensors or membrane switches, tactile push buttons were chosen because they offer better tactile feedback to voters, giving them a definite indication of successful activation as opposed to uncertainty of whether an input is registered (Herrnson *et al*., 2008).

Power supply subsystem will transform 220 V AC mains to 5 V DC which is regulated to be used with the Arduino. Although Arduino Uno might run off of USB power, polling station implementation needs autonomous power supply capacity not reliant on computing infrastructure which might either be inaccessible or unreliable. The implementation of battery back-up would be possible with simple adjustments so that it would continue to operate during power outages.

Some architectural decisions are informed by the consideration of security. Password authentication used in software will only permit the initiation of voting by authorized presiding officers who are required to type a predetermined password using a serial monitor before the system receives votes. Single-voteper-activation design implements over-voting prevention by software-enforced constraints. EEPROM storage: Non-volatile storage of votes that survive power outages – an important auditability need that allows post election verification. The lack of network connection is a form of convenience trading of the outcome of security decisions in the absence of network connectivity in order to achieve a reduced attack surface.

2.2 Power Supply Design and Circuit Implementation

A stable and reliable power supply is a basic prerequisite to EVM functionality because interruptions or changes in voltage may corrupt data, restart the system, or even make the system unusable (Scherz & Monk, 2013). The power supply subsystem will be used to convert ubiquitous 220 V AC mains to regulated 5V DC needed by Arduino Uno and other peripheral components. Fig. 2 demonstrates the power supply circuit diagram of cascaded stages of voltage transformation, rectification, filtering and regulation.

The design as illustrated in Fig. 2 utilizes the traditional linear regulation topology that is preferred due to its simplicity, the number of components, and the natural ability to reject noise (Scherz and Monk, 2013). The initial phase involves the use of a step-down transformer to reduce the 220 V AC to about 15 V AC. The choice of the transformer was based on the current consumption of Arduino Uno at a maximum 200 mA, plus more budget on LCD backlight (about 30 mA), resulting in a design current of 300 mA and a safety margin.

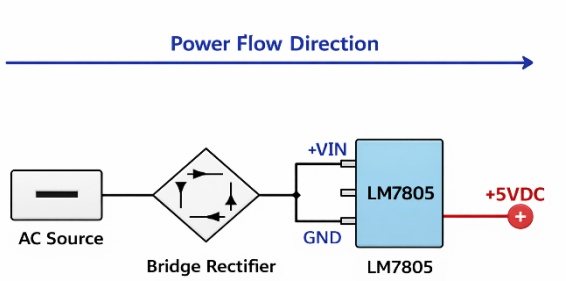

Fig. 2: Power supply circuit diagram to convert the 220 V AC to 5 V DC using step-down transformer, bridge rectifier, filter capacitors and an LM7805 voltage regulator. The design can accept the changes in the input voltage but still maintain a constant output of the necessary voltage to be used by the microcontroller

The bridge rectifier stage, which is accomplished in the form of four 1N4001 silicon diodes in a standard full-wave rectification arrangement, converts the AC voltage into pulsating DC (Horowitz and Hill, 2015). Full-wave rectification provides lower tolerances to half-wave rectification and minimizes the amplitude of the ripple voltage, which reduces the intensity of filtering systems (Scherz & Monk, 2013).

Filter capacitors even out pulsating DC output and minimize the ripple to the point that it can be further regulated. The selection of the capacitor value was done using the usual design procedure, which involved permissible ripple voltage and load current (Horowitz and Hill, 2015). Considering an input current of I = 300 mA in the microcontroller, output voltage V = 5V, rectifier frequency f = 100 Hz and tolerance to ripple, the minimum capacitance is 233.33 µF. A capacitor of 470 µF was picked with a sufficient filtering margin.

Voltage regulation is done by the LM7805 three-terminal linear regulator integrated circuit, which will provide a 5V constant output regardless of changes in input voltage or load current (Texas Instruments, 2016). The LM7805 is the industry-standard voltage regulator that is known to be simple, reliable and available everywhere. Further filtering capacitors (which are usually 100 nF ceramic capacitors located directly before regulator input and output connections) remove high-frequency noise and allow oscillations to undermine regulator stability (Texas Instruments, 2016).

The entirely connected circuit incorporates power, a microcontroller, a display, and input elements into seamless system running voting workflow. The finer circuit schematic in Fig. 3 depicts all the component interconnections and pin assignments.

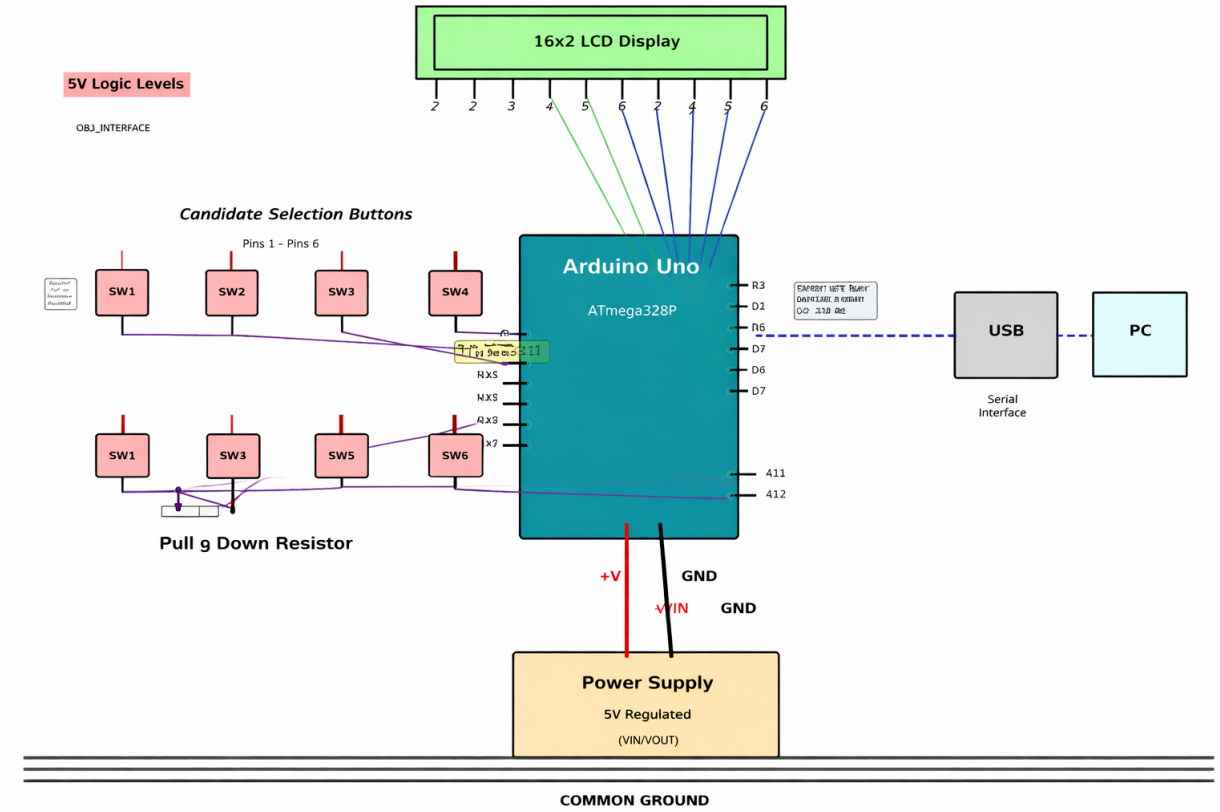

Fig. 3: Full circuit diagram of the Arduino Uno microcontroller connected to 16×2 LCD display, eight candidate selection push buttons and a power supply.

In the figure, Pin assignments LCD data pins (D4-D7) are connected to the Arduino digital pins 2-5, LCD control pins are connected to pin 7 and pin 6, and the push button is connected to the analogue input A0-A5 and the digital input 11-12 with the internal pull-up resistors turned on. Fig. 3 depicts that LCD is connected to the Arduino Uno with a 4-bit parallel interface that needs six microcontroller pins: four data lines (D4-D7) and two control signals (Register Select and Enable). Push button connections utilize internal pull-up resistors found on all the Arduino digital and analogue input pins, requiring no external circuitry to include discrete components of pull-up resistors (Atmel Corporation, 2015). The buttons will be connected with Arduino input pin and ground, turning on the internal pull-up resistor will keep the input to logic high (5V) when no button is pressed and the closure of the button will change the input to the ground (logic Low)

2.3 Software Development and Validation

The voting machine firmware was written in C/C++ in the Arduino IDE and it has an operational workflow that includes system

setup, presiding officer validation, voter validation, vote capture, tally accumulation and reporting the result. The software design focuses not only on functionality but also on being easily understandable and maintainable; that is, modularity in the separation of functions and extensive commentary. The workflow operational process, as shown in Fig. 4, is according to several separate states controlled by finite state machine in software.

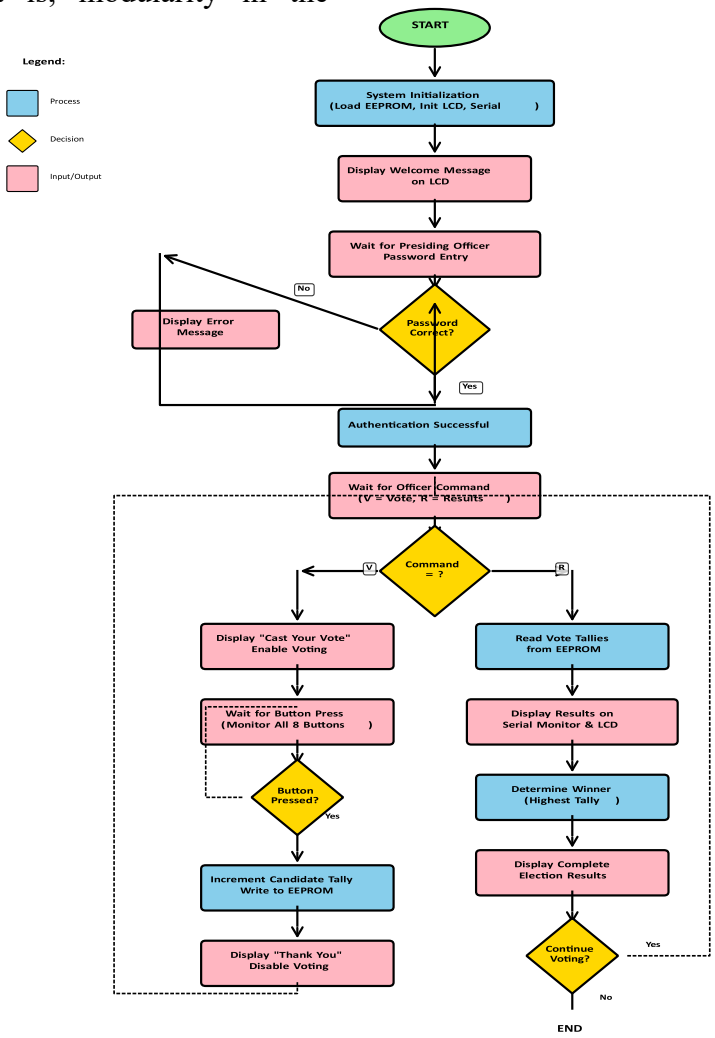

Fig. 4: Operational flowchart showing transitions between the states of the voting machine between power on initialize state to the presiding officer authentication state, voter authorization state, vote capture state, and result reporting state.

(The state machine design guarantees the correct sequencing of the electoral processes and avoids the unauthorized election voting and tampering with votes)

As it is depicted in Fig. 4, when power is switched on or when a system is restarted, it enters an initialize state where Arduino I/O pins are set up, LCD display is initialized with a welcome message, the EEPROM-stored vote counts are loaded into RAM variables, and serial communication with the computer of the presiding officer is established. Authentication of presiding officers needs to be done by typing in a predefined password (default: "VOTE") via the serial monitor interface. Firmware matches the characters received against the stored password string, and only on the correct match, it is changed to an authenticated state.

On authentication, commands may be issued by the presiding officer: 'V' to allow the voter to cast one vote or 'R' to get the cumulative result and display it. The 'V'

command changes to a voting-enabled state, where there is a prompt with instructions to vote by tapping a button with a favourite candidate. The firmware keeps a constant check on the eight button inputs by constant polling in the main program loop, which enforces the software debouncing by maintaining a continuous logic low level of at least 50 milliseconds before recognizing button press (Ganssle, 2008).

When the debounced button press is detected, the firmware updates the corresponding vote tally variable of the candidate and writes the new value to EEPROM, which will be read each time the device undergoes a boot cycle, displays a "thank you for voting" acknowledgement and returns to the idle state where a new presiding officer authorization is required to cast another vote. Such an access control implemented by software causes this single-vote-per-authorisation to ensure that over-voting is prevented. EEPROM write after vote registration means that each vote will be maintained even in the event of a power failure as soon as it is cast.

The command of the type "R" recalls the number of votes in EEPROM and sends it through the serial connection to the computer of the presiding officer, who displays it, enters it or sends it to the central counting center. The result format shows the identifiers of candidates and then the corresponding number of votes in human readable text format. EEPROM read access time is less than a second, which allows retrieving the result almost instantly in comparison with hours of counting the results manually.

Proteus Professional 8.2 software allowed performing virtual prototyping, and frequently shortened the development time to seconds, with software verification and debugging through virtual prototyping and physical hardware assembly eliminated by using both techniques, going through soldering, testing, and desoldering phases (Labcenter Electronics, 2018). Fig. 5 is a screenshot of proteus simulation environment with entire voting machine circuit in operation testing.

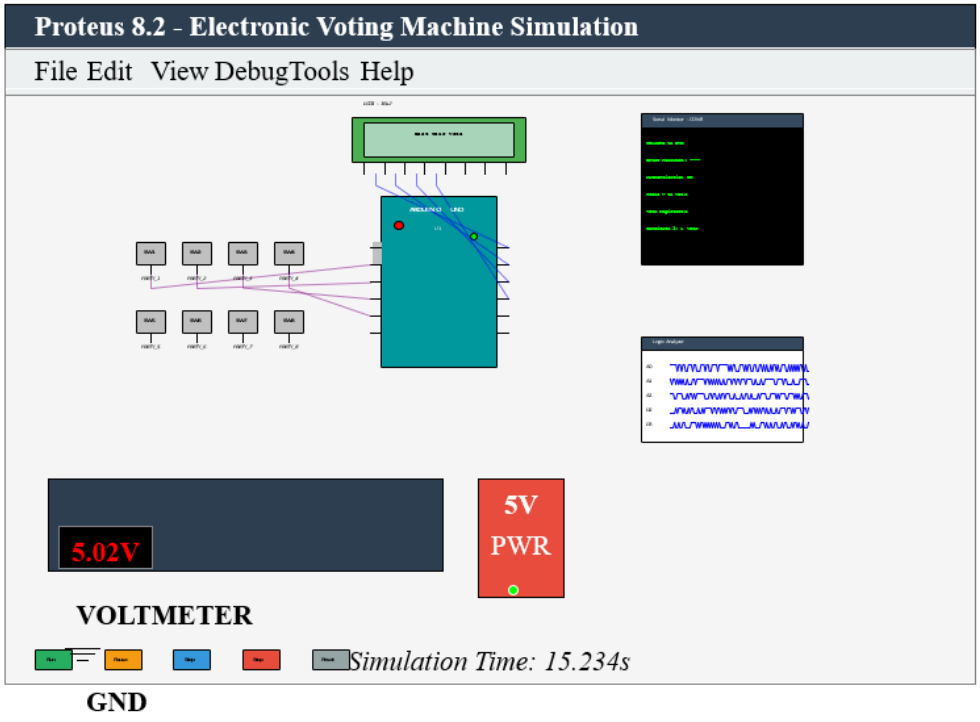

Fig. 5: The proteus simulation environment screenshot of the entire voting machine circuit in the functional test (Virtual instruments are used to monitor button inputs, LCD outputs and serial communication so that the logic of the firmware and interactions between components can be fully validated prior to physical prototyping)

Several operational conditions were exercised on simulation testing, such as regular voting sequences, attempted over-voting by repeatedly pressing the button, behavior on a power-on reset, and success and failure cases in password authentication, and the ability to retrieve the results given different tally states. These planned tests uncovered some serious firmware bugs in the preliminary stages of development that had been fixed before the hardware was developed.

After the successful validation of the simulation, construction of physical prototypes began. Fig. 6 demonstrates the prototype assembled, demonstrating key components and methods of construction.

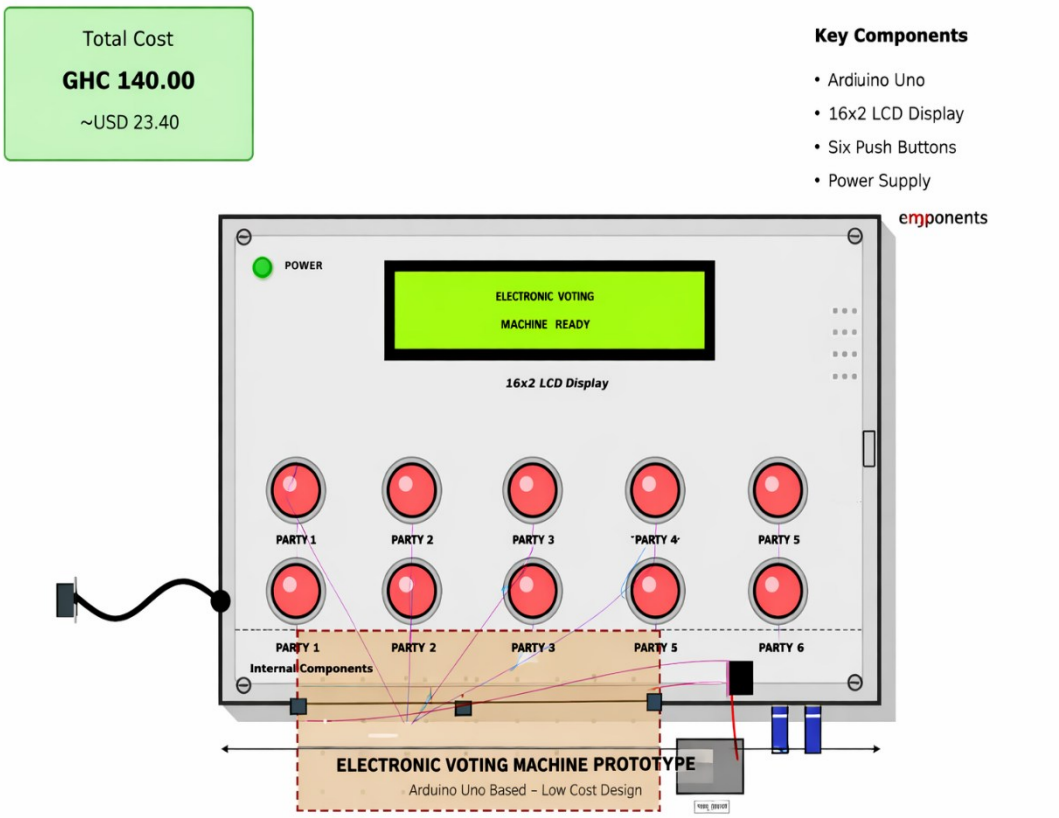

Fig. 6: Completed prototype of a voting machine consisting of Arduino Uno microcontroller, 16×2 LCD screen, eight candidate selection buttons in grid format, power supply electrical components on perforated board in protective case. (The construction itself proves the possibility of building the planned system with the help of the commonly accessible elements and the simplest methods of assembling)

The perforated board component mounting used in accordance with typical electronics assembly methods, as shown in Fig. 6 used through-hole soldering. LCD display is attached at the enclosure front behind a protective transparent cover. Push buttons stick out of labeled holes that can be felt and cannot be accessed internally. Interconnection of wires is colour-coded to enable problem diagnosis and maintenance. Modular construction is a method whereby it is possible to replace specific parts in case of failures.

Physical prototype testing confirmed functional operation over such parameters as vote capture accuracy with 100 + button press tests all with a 100 percent correct tally increase, LCD readability in controlled lighting conditions, stability of power supply with input voltage variation of 200-240 V AC, reliability of EEPROM data persistence with power cycling, and reliability of serial communication with multiple result retrieval cycles.

3.0 Results and Discussions

3.1 Performance Evaluation and Timing Analysis

Detailed performance testing was used to measure important operation measures such as time taken to process the votes, time taken to compile the results, accuracy of count and stability of the system, in the context of a realistic application. Table 1 presents the results of measured performance features of several test conditions with simulated voting sessions of different participants (voters) and different candidate distributions.

Table 1: Results of the voting machine prototype testing in the performance measures of various simulated election conditions. Values are mean values across 10 test sessions and the standard deviation is provided in cases where necessary

Performance Metric	Measured Value
Average time per vote (voter approach to completion)	2.4 ± 0.3 minutes
System response latency (button press to registration)	< 100 milliseconds
LCD display update latency	< 200 milliseconds
Vote counting accuracy (over 500 test votes)	100% (zero errors)
EEPROM write completion time	< 50 milliseconds
Complete result compilation and display time	0.8 ± 0.1 seconds
Power supply stability (output voltage variation)	± 20 mV
System availability (successful vote captures)	100% (zero failures)
Maximum continuous operation duration tested	8 hours

The average time per vote, as recorded in Table 1, time taken to approach the voter to completion and acknowledgement display, was 2.4 minutes in various test sessions. This is a good time when compared to the average times of paper ballot completion of 5- 10 minutes in conventional studies on voting systems (Herrnson *et al*., 2008), but direct comparison becomes complex due to the variation in ballot complexity.

Latency of system response – period between button press and internal vote registration – was less than 100 milliseconds in all tests, which is imperceptible to human voters who find vote capture to be instantaneous. This fast reaction is indicative of the lightweight computational load of incrementing integer variables and writing individual bytes to EEPROM, things very easily achievable with a 16 MHz ATmega328P. There was somewhat more LCD display update latency with an average value of less than 200 milliseconds, which is within the limits of interaction user interfaces where delays that are less than 300 milliseconds are not perceived by the user (Nielsen, 1993).

The accuracy of vote counting was 100% correct in 500+ test votes in a series of simulated elections. This is the best precision of a deterministic digital counting as compared to the manual tally counting errors-prone tally tabulation (Ansolabehere & Stewart, 2005). There were no errors encountered during the testing, which indicates that systematic bugs in the software that result in miscounting were also avoided during the simulation stage.

Average time to compile results and display took an average of 0.8 seconds once the presiding officer gave the command to display results on the serial monitor, and displayed results- this is a dramatic improvement over the time taken in hours or days in paper-based systems to count votes manually and aggregate the results (Alvarez and Hall, 2008). The availability of these results in near-instant time may potentially result in real-time monitoring of results by election observers, quick identification of suspicious anomalies,

and prompt distribution of results that will lower the tension linked to the excessive wait time. Its power supply performance was within the design requirements and delivered a stable output voltage over an input voltage variation of 200 V to 240 V AC (which is well within the operating tolerance of Arduino Uno) (Banzi and Shiloh, 2014). The longest continuous operation time was tested to be eight hours – more than a normal one day of voting time – without component failure, thermal problems or performance loss.

2.2 Economic Analysis and Cost Comparison

The economic feasibility is a crucial determinant of the technology adoption in resource-constrained electoral environments. Table 2 is a disaggregated component-by-component cost of the voting machine prototype, with prices showing the real cost of procuring the components through the Ghanaian electronic retailers in March 2017.

Table 2: Comprehensive costing of the components of the voting machine prototype. Ghanaian prices in Ghanaian Cedis (GH) in March 2017 at local electronic suppliers. US dollar equates at modern exchange rate of GH 4.43 against US$1.0.

Component	Cost (GH¤)	Cost (US$)
Arduino Uno R3 microcontroller board	50.00	11.29
16×2 character LCD display with backlight	25.00	5.64
Tactile push button switches (8 units)	16.00	3.61
Step-down transformer (220V/15V, 500mA)	18.00	4.06
1N4001 rectifier diodes (4 units)	2.00	0.45
Electrolytic capacitors (2200μF, 100nF)	4.00	0.90
LM7805 voltage regulator IC	3.50	0.79
Heat sink for voltage regulator	2.50	0.56
Perforated circuit board (veroboard)	8.00	1.81
Plastic enclosure and mounting hardware	6.00	1.35
Connecting wires, solder, and miscellaneous	5.00	1.13
Total Component Cost	140.00	31.60

The total cost of the components amounted to GH 140.00 (US 31.60) as explained in Table 2, making it notable in terms of affordability compared to commercial electronic voting machines that usually cost thousands to tens of thousands of dollars apiece (Gibson *et al*., 2016). Arduino Uno microcontroller board comprises the single most costly element at GH 50.00, which is about 36 percent of the overall hardware product, but this applies to retail markup on small-quantity buys and would reduce substantially on large-quantity procurements.

A long-term economic analysis of comparison with traditional paper-based voting shows strong economics despite the almost zero hardware cost of paper ballots. A Ghanaian voting station, which serves a population of 500 registered voters, needs about GH0.50 per ballot in terms of cost, which means that an election incurs GH balls give GH250.00 in cost (Electoral Commission of Ghana, 2016). Other expenses include the ballot boxes (GH fraction 100150), counting labour (GH fraction 300400), transmission of results (GH fraction 50100), and security (variable, but significant). Estimates on the cost of paper-based voting per-polling-station, and based on conservative estimates, it is in the range of GH 700- 1000 per election.

On the contrary, an electronic voting machine has a capital cost of GH140.00 and a very insignificant operating cost per election. Break-even on capital cost in

several electoral cycles, therefore takes about 2-3 elections. Later elections achieve net savings of GHferences GHferences - 1000 per polling station over paper-based methods, savings that translate into large amounts at the national level with thousands of polling stations.

Nevertheless, the total cost of ownership analysis should not ignore the caveats and other types of expenses that are not related to the initial acquisition of the hardware. Presiding officers and technical support personnel should be allocated some budget to train. Storage and transportation of voting machines between elections are expensive, similar to ballot box storage. This requires technical skills, which electoral commissions might require to develop or outsource to run software, as well as periodic updates of the firmware. Inventory prices on spare parts in case of component replacement will only contribute to small costs in the long run, but the construction in modules allows repair instead of implementing the replacement of the entire part.

1.4 Security Assessment and Comparative Analysis

Security assessment should take an adversarial stance, whereby attack vectors that potentially could be used by malicious actors to compromise the vote counts, breach ballot privacy, or tamper with the electoral process are systematically addressed (Kohno *et al*., 2004). Although the constructed system has multiple elements of security, the recognition of limitations and vulnerabilities should be regarded as critical intellectual honesty that guides proper deployment decision-making.

Authentication of presiding officers by passwords, though, offers a minimum level of security constraints against casual unauthorized access, is a low-level security boundary that can be penetrated by multiple attack styles. The transmits of passwords in cleartext over a serial connection and storing as plaintext string in firmware to enable shoulder-surfing attacks or firmware extraction attacks by adversaries who can access the Arduino with a physical device (Chattopadhyay *et al*., 2017). Stronger authentication may use one-time password tokens, challenge response protocols or biometric validation but each of these makes the system more complex and expensive.

A single-vote-per-authorisation system is useful in over-voting prevention, since the explicit action of a presiding officer is required to create a vote, and any further button press is ignored until another vote is registered. The implementation of this software-based access control presupposes that presiding officers will diligently perform their duties, that is, they will give one authorization to one verified voter, and that the polling station observers will control the absence of collusion. The mechanism effectively removes accidental or opportunistic over-voting, but fails to prevent systematic fraud through participation by presiding officers.

This standalone system, which does not have cryptographic protection of votes, is critical when it comes to physical security. Vote counts stored in EEPROM can be re-read or modified by anyone who can access the physical Arduino and programming device, allowing advanced attackers to manipulate the tallies provided the access is not supervised (Chattopadhyay *et al*., 2017). The mitigation options encompass tamper-evident seals on the enclosure to ensure that unauthorized access can be detected, constant presence of observers to ensure that there is no long period of unattended enclosures, and recording of serial numbers of deployed machines to ensure that the number of deployed machines matches the pre-election inventory.

Network isolation (intentional lack of WiFi, cellular, or wired network connection) is a measure of security strength that eliminates such remote attack vectors as man-in-the-middle result interception, denial-of-service attacks that disrupt transmission, and remote exploitation of network stack vulnerabilities (Springall *et al*., 2014). The physical serial

cable connectivity requirement to access results, although operationally inconvenient, greatly decreases attack surface to attackers that must access the physical polling station, which is far more difficult than remote network attacks.

The secrecy of ballots solely relies on the procedure controls and not cryptography. System does not have individual records on how specific voters cast their votes but instead records only aggregate vote counts per candidate so that it is impossible to track this backwards. But the temporal correlation attacks may even undermine secrecy when the observers observe the exact time when particular voters vote and match the timing with the order of registering a vote. Mitigation needs the visual privacy or a report of the results in batches of several votes before reporting.

In spite of these restrictions, a developed system security posture seems to be sufficient to be deployed in controlled polling stations with known and established observer presence and physical security measures similar to those used to protect traditional paper ballot systems. The target model that fits the current context is fundamentally different to the case in internet voting, where attackers have a free hand at remotely accessing the target. The use of polling stations limits attackers to physical presence, which poses a risk of detection, and restricts the scale of the attack.

Placing the developed system in relation to the previous research on Arduino-based voting machines sheds light on the contributions and continuity with previous research. The first attempt by Verma (2012) showed the basic appropriateness of Arduino Uno but used a microcontroller without EEPROM, which stored vote counts only in volatile SRAM that lost data when power was cut out, which is a crucial drawback that seriously undermines auditability. Recent studies help to fill this gap by using persistent vote storage as one of the main requirements.

The RFID-based smart EVM developed by Hussain *et al*. (2016) also included advanced voter authentication, automatic transfer of votes using GSM networks, and improved security due to the provision of cryptographic protection. Although these features are good add-ons, they also added huge costs to the systems by introducing RFID reader modules (US$20-40), GSM modems (US$3050) and RFID cards that had to be distributed to every qualified voter. Car distribution cost was not included in that total system cost would have been over US 150 per unit without card distribution cost-about five times present design budget.

Murali *et al*. (2016) created EVM which is a mixture of fingerprint biometric authentication and processing of microcontroller votes, which has a high level of voter verification virtually eliminating impersonation frauds. Nevertheless, there exist fingerprint sensors that can be aptly used to ensure the reliability of the sensor, but they cost US 3070 per sensor, which adds up to a significant amount of costs in the system. Minimalist approach in the current research takes the form of the absence of biometric authentication in the assumption that due to the deployment of the polling stations, the presiding officers are able to conduct the manual verification of the voters using the already available identity documents.

This study is also a unique one by being explicitly cost-minimizing prioritized and component-available without compromising the basic electoral functionality. Instead of trying to add all possible security improvements or convenience functions, the design is aimed at solid performance of the most critical voting business at a US$30-50 cost limit. This philosophy of minimalism is correlated with the principles of the appropriate choice of technology that suggest the development of solutions that correspond to the local resources and constraints instead of indiscriminate reception of technologically advanced and

contextually inappropriate methods (Schumacher, 1973).

4.0 Conclusion

The proposed study proves that cost-effective electronic voting machines based on Arduino microcontroller frameworks and off-the-shelf parts are technically and economically viable. The prototype developed manages to meet principal electoral functionality requirements such as proper vote capturing, tabulation of results in real-time, prevention of over-voting, and full auditability with a total hardware cost of less than US 32 per unit. Performance analysis indicates significant benefits over paper-based voting, such as voter processing time of less than three minutes, removal of counting error by manual counting with the use of deterministic digital tallying and immediate compilation of results with result compilation allowing real-time electoral monitoring. Economic analysis will break even with traditional systems after 2-3 electoral cycles, with the economy saving of costs in subsequent elections, accumulating to large quantities of money nationally. Security assessment also admits some critical weaknesses such as simple password authentication, a lack of cryptographic vote protection, and physical tampering by adversaries with unmonitored access to the machines, and these tradeoffs are accepted to permit high levels of affordability that may be used when a polling station is controlled with well-established physical security, and anyone is admitted. Its open-source design, permitting full publication of both schematics and firmware, gives the opportunity to independently verify it, adapt it locally, and build capacity unavailable to proprietary systems. Notable weaknesses that need further research are the absence of a built-in authenticator of voters, thus necessitating the use of outside authentication, unproven reliability in the long term in field deployment, lack of support for multiple languages and security threats to insider attacks. Proposed improvements would be to add low-cost biometric sensors since the cost of components has fallen, to add encrypted EEPROM storage, to add battery backup, to add creation of secure result transmission capability, and to perform field trials, creating empirical evidence regarding the challenges of operation and voter acceptance.

5.0 References

Alvarez, R. M., & Hall, T. E. (2008). *Electronic Elections: The Perils and Promises of Digital Democracy*. Princeton University Press. https://doi.org/10.1515/9781400837885

Ansolabehere, S., & Stewart III, C. (2005). Residual votes attributable to technology. *The Journal of Politics*, 67, 2, pp. 365-389. https://doi.org/10.1111/j.1468-2508.2005.00321.x

Atmel Corporation. (2015). *ATmega328P Datasheet: 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash*. http://ww1.microchip.com//en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P Datasheet.pdf

Banzi, M., & Shiloh, M. (2014). *Getting Started with Arduino* (3rd ed.). Maker Media, Inc.

Benaloh, J. (2006). Simple verifiable elections. In *Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop* (pp. 5-5). USENIX Association. https://doi.org/10.5555/1267609.1267614

Brunazo Filho, A., Cortiz, D., & Feitosa, D. (2015). Participant observation of the validation of a large scale electronic voting system. *Electronic Voting*, 205, pp. 113-124. https://doi.org/10.4230/OASIcs.EVOTE.2015.113

Caarls, S. (2010). *E-voting Handbook: Key Steps in the Implementation of E-enabled Elections*. Council of Europe Publishing.

Chattopadhyay, A., Lam, K. Y., & Tavva, Y. (2017). Autonomous vehicle: Security by design. *IEEE Transactions on Intelligent Transportation Systems*, 18, 11, pp. 3541-3549. https://doi.org/10.1109/TITS.2017.2713843

Chaum, D. (2004). Secret-ballot receipts: True voter-verifiable elections. *IEEE*

Security & Privacy, 2, 1, pp. 38-47. https://doi.org/10.1109/MSECP.2004.1264852

D'Ausilio, A. (2012). Arduino: A low-cost multipurpose lab equipment. *Behavior Research Methods*, 44, 2, pp. 305-313. https://doi.org/10.3758/s13428-011-0163-z

Edelstein, W. A., & Edelstein, A. D. (2010). Queuing and elections: Long lines, DREs and paper ballots. In *Proceedings of the 2010 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*. USENIX Association.

Electoral Commission of Ghana. (2016). *2016 Presidential and Parliamentary Elections Budget*. Accra: Electoral Commission.

Everett, S. P., Byrne, M. D., & Greene, K. K. (2008). Measuring the usability of paper ballots: Efficiency, effectiveness, and satisfaction. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 52, 24, pp. 1871-1875. https://doi.org/10.1177/154193120805202405

Feldman, A. J., Halderman, J. A., & Felten, E. W. (2007). Security analysis of the Diebold AccuVote-TS voting machine. In *Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop*. USENIX Association.

Ganssle, J. (2008). *The Art of Designing Embedded Systems* (2nd ed.). Newnes.

Gibb, A. (2014). *Building Open Source Hardware: DIY Manufacturing for Hackers and Makers*. Addison-Wesley Professional.

Gibson, J. P., Krimmer, R., Teague, V., & Pomares, J. (2016). A review of e-voting: The past, present and future. *Annales des Télécommunications*, 71, 7, 8, pp. 279-286. https://doi.org/10.1007/s12243-016-0525-8

Goggin, S. N., Byrne, M. D., & Gilbert, J. E. (2012). Post-election auditing: Effects of procedure and ballot type on manual counting accuracy, efficiency, and auditor satisfaction and confidence. *Election Law Journal*, 11, 1, pp. 36-51. https://doi.org/10.1089/elj.2010.0099

Gonggrijp, R., & Hengeveld, W. J. (2007). Studying the Nedap/Groenendaal ES3B voting computer: A computer security perspective. In *Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop*. USENIX Association.

Herrnson, P. S., Niemi, R. G., Hanmer, M. J., Bederson, B. B., Conrad, F. C., & Traugott,

M. (2008). *Voting Technology: The Not-So-Simple Act of Casting a Ballot*. Brookings Institution Press.

Horowitz, P., & Hill, W. (2015). *The Art of Electronics* (3rd ed.). Cambridge University Press.

Hussain, S. M., Ramaiah, C., Asuncion, R., Nizamuddin, S. A., & Veerabhadrappa, R. (2016). An RFID based smart EVM system for reducing electoral fraud. In *2016 International Conference on Reliability, Infocom Technologies and Optimization* (pp. 371-375). IEEE. https://doi.org/10.1109/ICRITO.2016.7784994

Jones, D. W. (2009). Paper vs. electronic voting records: An assessment. *IEEE Security & Privacy*, 7(1), 23-26. https://doi.org/10.1109/MSP.2009.3

Jones, D. W., & Simons, B. (2012). *Broken Ballots: Will Your Vote Count?* CSLI Publications.

Karlof, C., Sastry, N., & Wagner, D. (2005). Cryptographic voting protocols: A systems perspective. In *Proceedings of the 14th USENIX Security Symposium*, 14, pp. 33-50.

Kohno, T., Stubblefield, A., Rubin, A. D., & Wallach, D. S. (2004). Analysis of an electronic voting system. In *Proceedings of the 2004 IEEE Symposium on Security and Privacy* (pp. 27-40). IEEE. https://doi.org/10.1109/SECPRI.2004.1301313

Labcenter Electronics. (2018). *Proteus Design Suite Version 8 User Manual*. Labcenter Electronics Ltd.

Mercuri, R. (2002). A betterballotbox? *IEEE Spectrum*,39, 10, pp. 46-50. https://doi.org/10.1109/MSPEC.2002.1039937

Murali, S., Madhu, G. P., & Bojji, H. M. (2016). Design and implementation of secure electronic voting machine using biometric authentication. *International Journal of Computer Applications*, 145, 11, pp. 24-28. https://doi.org/10.5120/ijca2016910686

National Institute of Standards and Technology. (2007). *Voluntary Voting System Guidelines: Volume I, Version 1.0*. U.S. Department of Commerce.

Nielsen, J. (1993). Response times: The three important limits. In *Usability Engineering* (pp. 135-148). Academic Press.

Norris, P. (2014). *Why Electoral Integrity Matters*. Cambridge University Press. https://doi.org/10.1017/CBO9781107280861

Olembo, M. M., Kahlert, P. M., Kunz, T., & Volkamer, M. (2013). Securely voting from home. In *E-Voting and Identity: 4th International Conference, Vote-ID 2013* (pp. 207-224). Springer. https://doi.org/10.1007/978-3-642-39185-9_13

Runyan, N. (2007). Improving access to voting: A report on the technology for accessible voting systems. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* ,51, 5, pp. 827-831. . https://doi.org/10.1177/154193120705100514

Saltman, R. G. (2006). *The History and Politics of Voting Technology: In Quest of Integrity and Public Confidence*. Palgrave Macmillan. https://doi.org/10.1057/9780230288683

Scherz, P., & Monk, S. (2013). *Practical Electronics for Inventors* (3rd ed.). McGraw-Hill Education.

Schneier, B. (2004). What's wrong with electronic voting machines? *Open Democracy*, 9.

Schumacher, E. F. (1973). *Small Is Beautiful: Economics as if People Mattered*. Harper & Row.

Springall, D., Finkenauer, T., Durumeric, Z., Kitcat, J., Hursti, H., MacAlpine, M., & Halderman, J. A. (2014). Security analysis of the Estonian internet voting system. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (pp. 703-715). ACM. https://doi.org/10.1145/2660267.2660315

Teikari, P., Najjar, R. P., Malkki, H., Knoblauch, K., Dumortier, D., Gronfier, C., & Cooper, H. M. (2012). An inexpensive Arduino-based LED stimulator system for vision research. *Journal of Neuroscience Methods*, 211, 2, pp. 227-236. https://doi.org/10.1016/j.jneumeth.2012.09.012

Texas Instruments. (2016). *LM78XX series voltage regulators datasheet*. Retrieved from http://www.ti.com/lit/ds/symlink/lm7805.pdf

Valdés-Pérez, F. E., & Pallás-Areny, R. (2009). *Microcontrollers: Fundamentals and Applications with PIC*. CRC Press. https://doi.org/10.1201/9781420051223

Vassil, K., Solvak, M., Vinkel, P., Lumi, A. M., & Trechsel, A. H. (2016). The diffusion of internet voting: Usage patterns of internet voting in Estonia between 2005 and 2015. *Government Information Quarterly*, 33(3), 453-459. https://doi.org/10.1016/j.giq.2016.06.007

Verma, R. (2012). Arduino enabled electronic voting machine. *International Journal of Scientific and Engineering Research*, 3, 8, pp. 1-4.

Wilmshurst, T. (2009). *Designing Embedded Systems with PIC Microcontrollers: Principles and Applications* (2nd ed.). Newnes.

Wolchok, S., Wustrow, E., Isabel, D., & Halderman, J. A. (2012). Attacking the Washington, DC internet voting system. In *International Conference on Financial Cryptography and Data*

Security (pp. 114-128). Springer. https://doi.org/10.1007/978-3-642-32946-3_10

Declaration

Consent for publication

Not Applicable

Availability of data and materials

The publisher has the right to make the data public

Conflict of Interest

The authors declared no conflict of interest

Ethical Considerations

Not applicable

Competing interest

The authors report no conflict or competing interest

Funding

The author declared no source of funding

Author Contributions

All aspects of the work were carried out by the author