# **Voting Machine API Document**

#### ****Private Data Members****:

unsigned int holdingRegs[6]: Array to hold the vote counts for each candidate, total votes, and voting status.

* + holdingRegs[CANDIDATE\_1]: Votes for Candidate 1
  + holdingRegs[CANDIDATE\_2]: Votes for Candidate 2
  + holdingRegs[CANDIDATE\_3]: Votes for Candidate 3
  + holdingRegs[CANDIDATE\_4]: Votes for Candidate 4
  + holdingRegs[TOTAL\_VOTES]: Total number of votes
  + holdingRegs[VOTING\_STATUS]: Voting status (0 = active, 1 = completed)

const int pinDI1, pinDI2, pinDI3, pinDI4, pinDI8: GPIO pins for voting buttons and end voting button.

* + pinDI1: Vote for Candidate 1
  + pinDI2: Vote for Candidate 2
  + pinDI3: Vote for Candidate 3
  + pinDI4: Vote for Candidate 4
  + pinDI8: End Voting and Show Results

## **Public Member Functions:**

### **void begin()**

* **Description**: Initializes the voting machine by configuring the input pins for voting and setting up the system.
* **Parameters**: None
* **Return**: None

### **void checkVotes()**

* **Description**: Monitors the voting buttons and registers votes for candidates. It also checks for the "End Voting" button.
* **Parameters**: None
* **Return**: None

### **void configurePin(int pin)**

* **Description**: Configures the specified GPIO pin as an input with pull-up resistor.
* **Parameters**:
  + int pin: The GPIO pin to configure.
* **Return**: None

**bool debouncedInput(int pin, int index)**

* **Description**: Configures the specified GPIO pin as an input with pull-up resistor.
* **Parameters**:
  + int pin: The GPIO pin to check.
  + int index: The index of the button in the buttonPressed[] array to keep track of its state.
* **Return**: None

**Description**: Handles debouncing of button inputs. Ensures that each button press is registered only once, avoiding multiple votes for a single press.

**Parameters**:

* + - int pin: The GPIO pin to check.
    - int index: The index of the button in the buttonPressed[] array to keep track of its state.

**Return**: true if the button was pressed, false otherwise.

### *void registerVote(int candidate, const char candidateName)***\***

* **Description**: Registers a vote for the specified candidate and increments the total vote count.
* **Parameters**:
  + int candidate: The candidate number (0-3).
  + const char\* candidateName: The name of the candidate.
* **Return**: None

### **void displayResults()**

* **Description**: Displays the current voting results on the serial monitor.
* **Parameters**: None
* **Return**: None

### **void displayGuide()**

* **Description**: Displays a guide on how to use the voting machine (buttons and their actions).
* **Parameters**: None
* **Return**: None