

# Prepaid Meter Automation System - Project Build Report

## 1. Project Overview

This project automates keypad input on a prepaid energy meter using an ESP8266 Wi-Fi module, two stepper motors for X-Y movement, and a servo motor to tap keys. A mobile app sends a 5-digit token over Wi-Fi, which the system interprets and executes as button presses in a 3x4 keypad layout.

## 2. Hardware Components

- ESP8266 NodeMCU V2
- 2× L298N Motor Drivers
- 2× Bipolar Stepper Motors (e.g., 28BYJ-48 or NEMA 17)
- 1× SG90/standard servo motor
- 11.1V Li-ion battery (3× 3.7V cells)
- Prepaid energy meter with physical keypad
- Mechanical XY frame (sliders/rails/laser-cut or 3D printed parts)
- Common ground wiring and capacitors on L298N
- Optionally: capacitor across L298N VIN-GND for voltage stability

## 3. Software Structure

- ESP8266 firmware written in Arduino IDE using AccelStepper, Servo, and WiFi libraries
- MIT App Inventor app used as UI to send 5-digit token
- Web server hosted on ESP8266 listens for input at /submit?code=xxxxx
- Each digit maps to a 3x4 keypad grid and corresponding X-Y step positions
- Servo presses down at each digit location

## 4. System Building Process

Step 1: Wire two stepper motors to L298N drivers and connect them to ESP8266 GPIOs.

Step 2: Wire servo motor to a PWM GPIO (e.g., GPIO16) and power separately if needed.

Step 3: Connect L298N and ESP8266 to common ground and 12V power supply.

Step 4: Upload the test scripts to confirm stepper and servo operation.

Step 5: Upload final firmware and connect ESP8266 to your home Wi-Fi.

Step 6: Import the MIT App Inventor .aia project, build the app, and install on phone.

Step 7: Enter the 5-digit token in the app and verify accurate button pressing.

Step 8: Adjust STEP\_X\_SPACING and STEP\_Y\_SPACING constants to match your keypad size.

## **5. Functional Flow**

1. App connects to ESP8266 over Wi-Fi
2. User enters 5-digit code
3. App sends request to ESP8266: /submit?code=12345
4. ESP maps each digit to a grid row/column
5. Converts grid position to motor steps
6. Moves to position and triggers servo tap
7. Remembers last position for optimized next move
8. Repeats for all digits

## **6. How to Replicate This Project**

- Build a 3x4 keypad grid to match your meter layout
- Measure spacing between buttons (in mm) and convert to stepper motor steps
- Update STEP\_X\_SPACING and STEP\_Y\_SPACING values accordingly
- Ensure all components share a common ground
- Set your Wi-Fi SSID/password in the code
- Reuse the provided MIT App Inventor project or customize it
- Upload code, power system, and verify input handling and movement accuracy