```
#include "U8g2lib.h"

// ==========================
// VARIABLE DEFINITIONS
// ==========================
#define SCK 13   // Serial Clock (SPI)
#define SDA 11   // MOSI (SPI)
#define CS 10    // Chip Select
#define DC 9     // Data/Command
#define Reset 8  // Reset Pin

const int potPinPlayers = A0;
const int potPinCards = A1;
const int dirPin = 4;
const int stepPin = 3;
const float stepAngle = 1.8;
const float stepsPerRevolution = 360.0 / stepAngle;
int moottori_1 = 6;
int moottori_2 = 5;
const float driveGearDiameter = 12.0;
const float drivenGearDiameter = 36.0;
const float gearRatio = drivenGearDiameter / driveGearDiameter;
int button = 2;

volatile bool interruptTriggered = false;
bool settingsLocked = false;
bool dealingInProgress = false;
int lockedPlayers = 1;
int lockedCards = 1;
// OLED setup
U8G2_SH1106_128X64_NONAME_1_4W_SW_SPI u8g2(U8G2_R0, SCK, SDA, CS, DC, Reset);

// Function that stabilizes potentiometer values
int stableAnalogRead(int pin) {
    int sum = 0;
    for (int i = 0; i < 5; i++) {
        sum += analogRead(pin);
        delay(10);
    }
    return sum / 5;
}

int calculate_players() {
    if (settingsLocked) return lockedPlayers;
    int potValue = stableAnalogRead(potPinPlayers);
    Serial.println(stableAnalogRead(potPinPlayers));
    return map(potValue, 0, 1023, 1, 8);
}

int calculate_cards() {
    if (settingsLocked) return lockedCards;
    int potValue = stableAnalogRead(potPinCards);
    int players = calculate_players();
    int maxCardsPerPlayer = 52 / players;
```

```
    return map(potValue, 0, 1023, 1, maxCardsPerPlayer);
}

void rotateStepperByPlayers() {
    int players = calculate_players();
    float angle = 360.0 / players;
    float adjustedAngle = angle * gearRatio;
    int steps = (adjustedAngle / stepAngle);
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < abs(steps); i++) {
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(4000);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(4000);
    }
}

void fullRotation() {
    int players = calculate_players();
    float stepAnglePerPlayer = 360.0 / players;
    float rotatedAngle = 0;
    dealingInProgress = true;

    while (rotatedAngle < 360) {
        rotateStepperByPlayers();
        rotatedAngle += stepAnglePerPlayer;
        digitalWrite(moottori_1, LOW);
        analogWrite(moottori_2, 240);
        delay(180);
        analogWrite(moottori_1, 180);
        digitalWrite(moottori_2, LOW);
        delay(180);
        digitalWrite(moottori_1, LOW);
        digitalWrite(moottori_2, LOW);
        delay(250);
    }

}

void triggerFullRotation() {
     if (dealingInProgress) return; // Prevent actions from the button while dealing is
in progress.
    if (!settingsLocked) {
        lockedPlayers = calculate_players();
        lockedCards = calculate_cards();
        settingsLocked = true;
    }
    interruptTriggered = true;
}

void setup() {
    Serial.begin(9600);
    u8g2.begin();
    pinMode(potPinPlayers, INPUT);
```

```
    pinMode(potPinCards, INPUT);
    pinMode(dirPin, OUTPUT);
    pinMode(stepPin, OUTPUT);
    pinMode(moottori_1, OUTPUT);
    pinMode(moottori_2, OUTPUT);
    pinMode(button, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(button), triggerFullRotation, FALLING);
    u8g2.clearBuffer();
    u8g2.setFont(u8g2_font_ncenB08_tr);
    u8g2.drawStr(10, 30, "OLED Test");
    u8g2.sendBuffer();
    delay(2000);
}

void loop() {
    u8g2.firstPage();
    do {
        u8g2.setFont(u8g2_font_ncenB08_tr);
        if (settingsLocked) {
            u8g2.drawStr(10, 20, "Players:");
            u8g2.setCursor(100, 20);
            u8g2.print(lockedPlayers);
            u8g2.drawStr(10, 40, "Cards/Player:");
            u8g2.setCursor(100, 40);
            u8g2.print(lockedCards);
        } else {
            int players = calculate_players();
            int cards = calculate_cards();
            u8g2.drawStr(10, 20, "Players:");
            u8g2.setCursor(100, 20);
            u8g2.print(players);
            u8g2.drawStr(10, 40, "Cards/Player:");
            u8g2.setCursor(100, 40);
            u8g2.print(cards);
        }
    } while (u8g2.nextPage());

    if (interruptTriggered) {
        for (int i = 0; i < lockedCards; i++) {
            fullRotation();
        }
        interruptTriggered = false;
        settingsLocked = false;
        dealingInProgress = false;
    }
    delay(1000);
}
```