

TTP 229 Keypad and nano Arduino password based digital lock

```
#define SCL 2          // Define the SCL pin
#define SDO 3          // Define the SDO pin
#define LOCK_PIN 4     // Define the lock control pin

// Define the correct password
String password = "1234"; // Password to unlock the lock
String inputPassword = ""; // String to store the input password

byte Key; // To store the key pressed

// Function to read the key pressed from TTP229 keypad
byte Read_TTP229_Keypad() {
    byte Num;
    byte Key_State = 0;
    for (Num = 1; Num <= 16; Num++) {
        digitalWrite(SCL, LOW);
        if (!digitalRead(SDO)) {
            Key_State = Num;
        }
        digitalWrite(SCL, HIGH);
    }
    return Key_State;
}

void setup() {
    Serial.begin(9600); // Start serial communication
    pinMode(SCL, OUTPUT); // Set SCL pin as output
    pinMode(SDO, INPUT); // Set SDO pin as input
    pinMode(LOCK_PIN, OUTPUT); // Set the lock pin as output
    lock(); // Start with the lock engaged
}

void loop() {
    Key = Read_TTP229_Keypad(); // Read the key pressed from the TTP229 keypad

    if (Key) { // If a key is pressed
        Serial.print("Key pressed: ");
        Serial.println(Key);
        delay(500); // Small delay for debouncing the key press

        // Map the key number to a corresponding character (1-9, 0, C, D)
        char keyChar = mapKeyToChar(Key);
```

```

if (keyChar != '\0') {
    // Only add numbers and letters to the password
    if (keyChar != 'D' && keyChar != 'C') {
        inputPassword += keyChar; // Append the pressed key to the input password
    }
    Serial.print("Current input password: ");
    Serial.println(inputPassword);
}

// Check for 'Enter' key (D) to validate password
if (Key == 16) { // Assuming 'D' is mapped to key 16
    if (inputPassword == password) {
        Serial.println("Password correct! Unlocking...");
        unlock();
        delay(30000); // Keep the lock open for 5 seconds
        lock();
    } else {
        Serial.println("Incorrect password.");
    }
    inputPassword = ""; // Clear the input password
}

// Check for 'Clear' key (C) to reset the input password
if (Key == 15) { // Assuming 'C' is mapped to key 15
    inputPassword = ""; // Clear the input password
    Serial.println("Input password cleared.");
}
}

delay(200); // Small delay for key debounce
}

// Function to map key number to corresponding character
char mapKeyToChar(byte keyNum) {
    switch (keyNum) {
        case 1: return '1';
        case 2: return '2';
        case 3: return '3';
        case 4: return '4';
        case 5: return '5';
        case 6: return '6';
        case 7: return '7';
        case 8: return '8';
        case 9: return '9';
        case 10: return 'A'; // Optional (can be used for future use)
        case 11: return '0';
        case 12: return 'B'; // Optional (can be used for future use)
        case 13: return 'C'; // Clear key
        case 14: return 'D'; // Enter key
        case 15: return 'C'; // Clear key (again for clarity)
    }
}

```

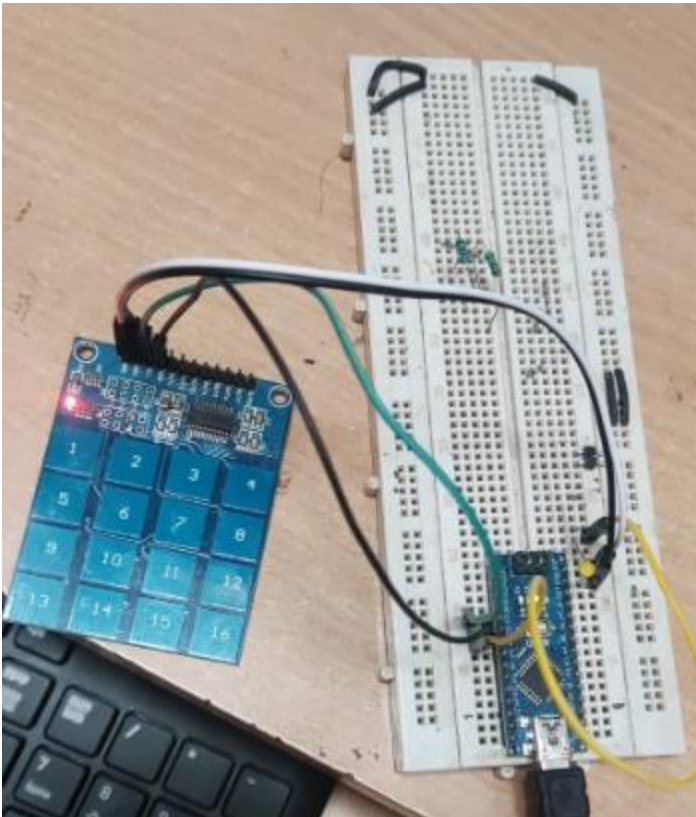
```

    case 16: return 'D'; // Enter key (again for clarity)
    default: return '\0'; // If no valid key is pressed
}
}

// Function to lock the electromagnetic lock
void lock() {
    digitalWrite(LOCK_PIN, LOW); // Lock engaged (LOW to lock, HIGH to unlock)
    Serial.println("Lock engaged.");
}

// Function to unlock the electromagnetic lock
void unlock() {
    digitalWrite(LOCK_PIN, HIGH); // Unlock the lock (HIGH to unlock, LOW to lock)
    Serial.println("Lock disengaged.");
}

```



Step 1: Verify Wiring • TTP229 SDA → Arduino D3 • TTP229 SCL → Arduino D2 • TTP229 VCC → Arduino 5V • TTP229 GND → Arduino GND

