HANNIBAL HERRERA - TECHNICAL PORTFOLIO

Technology Operations Manager - Universal Horror Unleashed
Demonstrating Cross-Platform Expertise & Automation Skills

🚦 FEATURED PROJECT: SAFETY & SCENE CONTROL SYSTEM

Perfect example of automation skills applicable to UHU special effects systems

Project Overview

Developed a multi-platform safety and scene control system demonstrating:

- **Cross-platform compatibility** (Windows, macOS, Linux)
- **G-code/M-code programming** for hardware control
- **Automation sequences** for entertainment applications
- **Real-time system control** capabilities

Technical Implementation

🔴 **RED LIGHT SEQUENCE (Emergency Stop)**

```
; Emergency Stop Protocol - Critical for UHU safety systems
M112 ; Emergency stop all operations
G28 ; Home all positions
M106 S0 ; Turn off all effects
G4 P5000 ; Hold for 5 seconds
M117 "EMERGENCY STOP ACTIVE" ; Display message
```

🟡 **YELLOW LIGHT SEQUENCE (Warning/Transition)**

```
; Warning Transition - Perfect for UHU scene changes
M106 S128 ; Set warning effects to 50%
G4 P2000 ; Hold warning for 2 seconds
M117 "SCENE TRANSITION" ; Display status
G1 F3000 ; Set transition speed
```

🟢 **GREEN LIGHT SEQUENCE (Go/Active)**

```
; Full Operations Active - UHU show running
M106 S255 ; Full effects power
G1 X100 Y100 Z50 F6000 ; Move to active position
M117 "HORROR UNLEASHED" ; Display show status
G4 P10000 ; Run full sequence 10 seconds
```

💻 CROSS-PLATFORM EXPERTISE

Windows Environment

```
# PowerShell automation script for Windows systems
# Perfect for Universal's Windows-based POS and management systems
```

```powershell
# Traffic Light Controller - Windows Implementation
function Start-TrafficController {
    param(
        [string]$Mode = "Auto",
        [int]$CycleDuration = 30
    )

    Write-Host "🚦 Starting Traffic Controller on Windows..." -ForegroundColor Yellow

    # Initialize hardware connections
    $SerialPort = New-Object System.IO.Ports.SerialPort("COM3", 9600)
    $SerialPort.Open()

    # Main control loop
    while ($true) {
        # RED PHASE - Emergency protocols
        Send-GCode "M106 S255" # Activate red light
        Write-Host "🔴 RED - Emergency Stop Active" -ForegroundColor Red
        Start-Sleep -Seconds 10

        # YELLOW PHASE - Transition warning
        Send-GCode "M106 S128" # Activate yellow light
        Write-Host "🟡 YELLOW - Transition Warning" -ForegroundColor Yellow
        Start-Sleep -Seconds 3

        # GREEN PHASE - Full operations
        Send-GCode "M106 S0; M107" # Activate green light
        Write-Host "🟢 GREEN - Horror Unleashed!" -ForegroundColor Green
        Start-Sleep -Seconds 17
    }
}

# Hardware monitoring for UHU reliability
function Monitor-SystemHealth {
    Get-Counter "\Processor(_Total)\% Processor Time" -Continuous
    Get-EventLog -LogName System -Newest 10 | Where-Object {$_.EntryType -eq "Error"}
}
```

macOS Environment

```bash
#!/bin/bash
# macOS automation script using native tools
# Demonstrates cross-platform compatibility for diverse UHU systems

# Traffic Light Controller - macOS Implementation
function traffic_controller_mac() {
    echo "🚦 Starting Traffic Controller on macOS..."

    # Use macOS native serial communication
    if [[ -e /dev/cu.usbmodem* ]]; then
        DEVICE=$(ls /dev/cu.usbmodem* | head -n1)
        echo "📡 Connected to: $DEVICE"
```

```
        # Configure serial settings for hardware control
        stty -f $DEVICE 9600 cs8 -cstopb -parity

        while true; do
            # RED PHASE - Critical safety protocols
            echo "M106 S255" > $DEVICE  # Full red activation
            echo "🔴 RED - Emergency Safety Mode"
            osascript -e 'display notification "Emergency Mode Active" with title "UHU Safety
System"'
            sleep 10

            # YELLOW PHASE - Scene transition
            echo "M106 S128" > $DEVICE  # Warning level
            echo "🟡 YELLOW - Scene Transition"
            afplay /System/Library/Sounds/Ping.aiff  # Audio feedback
            sleep 3

            # GREEN PHASE - Show active
            echo "M106 S0" > $DEVICE    # Green activation
            echo "🟢 GREEN - Horror Experience Active!"
            osascript -e 'display notification "Horror Unleashed!" with title "UHU Show Active"'
            sleep 17
        done
    else
        echo "❌ No hardware connection found"
        exit 1
    fi
}

# System monitoring with macOS tools
function monitor_mac_systems() {
    # CPU and memory monitoring
    top -l 1 | grep "CPU usage"
    vm_stat | grep "Pages free"

    # Network connectivity check for UHU systems
    ping -c 1 area15.com &> /dev/null && echo "✅ Network OK" || echo "❌ Network Issue"
}
```

Linux Environment

```
#!/bin/bash
# Linux automation script for robust server environments
# Perfect for UHU backend systems and reliability

# Traffic Light Controller - Linux Implementation
function traffic_controller_linux() {
    echo "🚦 Starting Traffic Controller on Linux..."

    # Detect available serial devices
    for device in /dev/ttyUSB* /dev/ttyACM*; do
        if [[ -e $device ]]; then
```

```
            echo "📡 Found device: $device"

            # Configure device for G-code communication
            stty -F $device 9600 cs8 -cstopb -parity raw

            # Main automation loop
            while true; do
                # RED PHASE - System safety first
                echo "G28" > $device          # Home position
                echo "M112" > $device         # Emergency stop ready
                echo "M106 S255" > $device    # Red light full power
                echo "🔴 RED - Safety Protocol Active"
                logger "UHU Safety: Red light phase activated"
                sleep 10

                # YELLOW PHASE - Transition state
                echo "M106 S128" > $device    # Yellow warning level
                echo "🟡 YELLOW - Preparing Scene Change"
                logger "UHU Operations: Scene transition initiated"
                sleep 3

                # GREEN PHASE - Full show operations
                echo "M106 S0" > $device      # Green operations
                echo "G1 F6000" > $device     # High-speed operations
                echo "🟢 GREEN - HORROR UNLEASHED!"
                logger "UHU Show: Full horror experience active"
                sleep 17
            done
        fi
    done

    echo "❌ No hardware devices found"
    exit 1
}

# Advanced Linux system monitoring for UHU reliability
function monitor_linux_systems() {
    # System resource monitoring
    echo "=== SYSTEM HEALTH CHECK ==="
    echo "CPU Usage: $(cat /proc/loadavg)"
    echo "Memory: $(free -h | grep Mem)"
    echo "Disk Space: $(df -h / | tail -1)"

    # Network connectivity for show systems
    echo "=== NETWORK STATUS ==="
    ping -c 1 8.8.8.8 &> /dev/null && echo "✅ Internet: Connected" || echo "❌ Internet: Failed"

    # Hardware device status
    echo "=== HARDWARE STATUS ==="
    lsusb | grep -i "serial\|arduino\|usb" | head -5

    # System log monitoring for issues
    echo "=== RECENT SYSTEM EVENTS ==="
```

```
    journalctl -p err -n 5 --no-pager
}
```

🎬 ENTERTAINMENT INDUSTRY APPLICATIONS

UHU Special Effects Integration

This traffic light controller demonstrates skills directly applicable to Universal Horror Unleashed:

🎃 Horror Scene Management

```
; Halloween Scene Controller for UHU
; Manages lighting, fog, and animatronics

M117 "Preparing Scare Sequence"
G28 ; Reset all positions
M106 S0 ; Lights off - darkness

; Build suspense phase
G4 P2000 ; Wait 2 seconds in darkness
M106 S64 ; Dim ambient lighting
G1 Z10 F1000 ; Slowly raise animatronic

; SCARE ACTIVATION!
M106 S255 ; FULL STROBE LIGHTS!
G1 Z100 F6000 ; RAPID ANIMATRONIC MOVEMENT!
M42 P13 S255 ; Activate fog machine
G4 P3000 ; Hold scare for 3 seconds

; Recovery phase
M106 S128 ; Reduce to mood lighting
G1 Z50 F2000 ; Lower animatronic slowly
M42 P13 S0 ; Stop fog
M117 "Scare Complete - Reset"
```

🚨 Emergency Safety Protocols

```
; Emergency shutdown for guest safety
; Critical for UHU operations

M112 ; IMMEDIATE STOP ALL MOTION
M107 ; Turn off all lighting effects
M42 P13 S0 ; Stop fog machines
M42 P14 S0 ; Stop audio effects
G28 ; Return all elements to safe positions
M117 "EMERGENCY STOP - GUEST SAFETY"

; Safety check sequence
G4 P5000 ; Hold all systems stopped
M114 ; Report current positions
M117 "System Safe - Awaiting Reset"
```

🔧 HARDWARE INTEGRATION EXPERTISE

Supported Platforms & Devices

- **Arduino Uno/Mega** - Perfect for UHU animatronics
- **Raspberry Pi** - Excellent for show control systems
- **Serial Communication** - Essential for equipment control
- **USB/Bluetooth** - Wireless show management
- **Network Integration** - Remote monitoring capabilities

Real-Time Control Capabilities

- **Microsecond precision** timing for synchronized effects
- **Multi-device coordination** for complex scenes
- **Emergency stop protocols** for guest safety
- **Remote monitoring** and diagnostics
- **Cross-platform compatibility** for diverse UHU systems

🎯 UNIVERSAL HORROR UNLEASHED APPLICATIONS

Direct Skill Applications

1. 🎭 **Animatronic Control** - G-code programming for creature movements
2. 💡 **Lighting Sequences** - Automated strobe and mood lighting
3. 🌫️ **Fog Machine Timing** - Synchronized atmospheric effects
4. 🔊 **Audio Coordination** - Timed sound effect triggers
5. 🚨 **Safety Systems** - Emergency stop and guest protection
6. 📊 **System Monitoring** - Real-time diagnostics and alerts

Operational Benefits for UHU

- **Reliability**: Cross-platform expertise ensures system stability
- **Safety**: Military-grade emergency protocols protect guests
- **Innovation**: Creative automation enhances horror experiences
- **Efficiency**: Streamlined operations reduce downtime
- **Scalability**: Skills applicable across all UHU technology systems

---

🏆 PORTFOLIO HIGHLIGHTS

Why This Matters for UHU

✅ **Automation Expertise** - Essential for special effects coordination
✅ **Cross-Platform Skills** - Works with any UHU system architecture
✅ **Safety Focus** - Military training ensures guest protection protocols
✅ **Real-Time Control** - Perfect for live entertainment requirements
✅ **Creative Problem Solving** - Innovative solutions for unique challenges
✅ **Emergency Response** - Crisis management skills for system failures

Technical Competencies Demonstrated

- **G-code/M-code Programming** ⭐ Advanced level
- **Multi-Platform Development** ⭐ Expert level
- **Hardware Integration** ⭐ Professional level
- **System Automation** ⭐ Advanced level
- **Safety Protocols** ⭐ Military-grade level

- **Real-Time Systems** ⭐ Expert level

🎓 PROFESSIONAL CERTIFICATIONS

View All Certificates: https://skynetadmin.github.io/Universal-Studios-Portfolio/certificates/

✅ **Google IT Support Professional Certificate** (Verification: B5Q9RBNG6A7F)
✅ **Google IT Automation with Python Certificate** (Verification: KT3ZH8U95EKP)
✅ **Google Technical Support Fundamentals** (Verification: N4NR9D7ABU22)
✅ **Google Operating Systems and You: Becoming a Power User** (Verification: NBQTPBUJXEY7)
✅ **Professional Technology Operations Training** (Summary Report Available)

All certificates verified and available for employer review via secure link above

---

📞 CONTACT & DEMONSTRATION

Hannibal Herrera
📧 herrera.hannibal84@gmail.com
📱 702-626-7678
📍 Las Vegas, NV (5.5 miles from Area 15)

🎬 **Live Demonstration Available**
Ready to showcase Universal Horror Unleashed automation systems during interview

💻 **Interactive Portfolio**
🌐 **Website:** https://skynetadmin.github.io/Universal-Studios-Portfolio/website/

Portfolio Demonstrations:

- 👹 **Animatronic Witch Controller**: Arduino Mega 2560 + RAMPS 1.4 with G-Code programming
- 🚦 **Safety & Scene Control System**: Emergency protocols and industrial automation integration
- 🌫️ **Fog Machine Choreography**: DMX512 protocol with multi-zone deployment and RGB lighting
- 📊 **Technology Operations Management Dashboard**: Executive-level oversight with KPIs and team management
- 🐍 **Python Automation Scripts**: Docker orchestration, serial communication, and system monitoring

Complete source code and documentation available upon request

---