**Arduino and NI MyDAQ: Power Integrity, Digital I/O, PWM, and Serial Communication Test**

**Project Overview**

This project demonstrates my ability to integrate an **Arduino Uno** with **NI MyDAQ** hardware, automated through **LabVIEW**, to perform sequential electronic validation tests.  
It showcases a complete and realistic test sequence involving:

* **Power Rail Integrity Testing (5V and 3.3V)**
* **Digital I/O Communication and Response Time Measurement**
* **PWM Signal Generation and Monitoring**
* **Serial Communication Validation**

All communication between LabVIEW, NI MyDAQ, and Arduino was successfully achieved, overcoming multiple hardware and software challenges.

**Project Description**

The system automatically performs the following tests in sequence:

1. **Power Integrity Test (5V and 3.3V Rails)**
   * **5V Rail Test:**
     + The Arduino’s 5V rail is measured using **NI MyDAQ Analog Input AI0**.
   * **3.3V Rail Test:**
     + The Arduino’s 3.3V rail is measured using **NI MyDAQ Analog Input AI1**.
     + A **5V relay module** is used to connect/disconnect the 3.3V line:
       - The 3.3V line is connected to the **Normally Open (NO)** contact of the relay.
       - During the 3.3V rail test, Arduino **closes** the relay (writing HIGH to the relay control).
       - After the 3.3V rail test, Arduino **opens** the relay (writing LOW) to disconnect the 3.3V rail, freeing AI1 for the PWM tests.
   * **Noise Reduction:**
     + A **capacitor** was connected across the AI1 positive and negative terminals to stabilize readings during analog measurements.
2. **Digital I/O Test**
   * LabVIEW (via NI MyDAQ) sends a digital pulse to the Arduino.
   * Arduino detects the incoming pulse and sends a return pulse back to MyDAQ.
   * The **response time** between sending and receiving the pulse is measured.
   * Arduino’s onboard LED provides a visual confirmation during this test.
3. **PWM Signal Testing**
   * Arduino generates PWM signals with duty cycles of:
     + **30%**
     + **60%**
     + **90%**
   * These PWM signals are monitored through **NI MyDAQ Analog Input AI1** (after the 3.3V rail is disconnected).
4. **Serial Communication Test**
   * LabVIEW sends a 19-character string via serial communication to Arduino.
   * Arduino reads and echoes the string back to confirm serial communication reliability.

**Hardware Used**

* Arduino Uno
* NI MyDAQ
* Jumper wires
* 5V Relay Module (for rail switching)
* Capacitor (across AI1 terminals to improve analog signal stability)
* Laptop running LabVIEW 2019 or later

**Software Used**

* LabVIEW (for test automation and NI MyDAQ control)
* Arduino IDE (for firmware development)

**Key Challenges Overcome**

* **Shared Analog Channel Management:**  
  Designed a switching system using a relay to allow AI1 to serve both the 3.3V rail measurement and PWM capture without conflict.
* **Reliable Serial Communication Handling:**  
  Ensured that serial transmission and echo were properly synchronized without disrupting the test flow.
* **Precise Digital I/O Response Measurement:**  
  Implemented a mechanism to accurately measure response times between Arduino and LabVIEW.
* **State Machine Logic:**  
  Built a robust Labview state machine to correctly sequence the tests and synchronize LabVIEW triggers and readings with Arduino.
* **Hardware Integration:**  
  Successfully interfaced NI MyDAQ's analog and digital capabilities with Arduino's logic and voltage levels.
* **Noise Minimization:**  
  Added a capacitor across AI1 terminals to reduce measurement noise during analog readings.

**What This Project Demonstrates**

* Integration of **Arduino** and **NI MyDAQ** into a **LabVIEW-driven automated test system**.
* Application of **state machine programming** in embedded firmware (Arduino).
* Handling of **mixed signal acquisition** (digital + analog) within the same test flow.
* Basic **power rail verification** techniques for hardware validation.
* **Relay control** for hardware resource optimization (channel sharing).
* Overcoming practical issues in **serial communication** and **real-time testing**.