Capstone: Relay Analyzer

Kevin McIntosh & Matthew Hengeveld

Electronic Systems Engineering - Conestoga College

In This Presentation

1. Project Description
2. System Requirements
3. System Design
4. Test Methodology
5. Results and Analysis
6. Status and Next Steps

Project Description

- The “Relay Analyzer”
- Automatically tests relays to determine characteristics
  - Coil voltage
  - Coil current
  - Coil resistance
  - Activation and deactivation times
  - Bounce characteristics
- Tracks characteristics over time and use
- Determine any change in characteristics and performance
- Can determine the number of cycles before failure

Relay Analyzer

- Many different types of relays
- Each relay has its own characteristics
- Relay manufacturers only put certain, limited characteristics on datasheets
- Relay Analyzer is a tool to automatically test a relay to determine all its characteristics, including how those characteristics change with time and use
- Allows the design engineer to spend less time choosing a relay, and testing them to fit their application

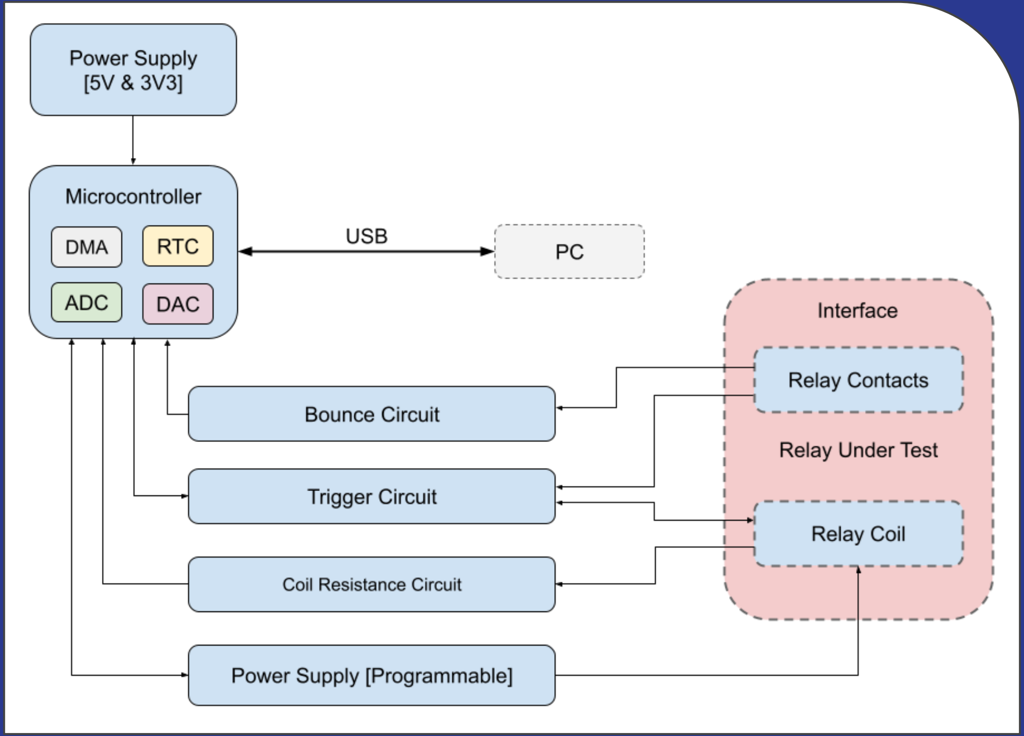
Project Requirements

- Measure the activation and deactivation times of a relay
- Measure and characterize the switch bounce
- Adjust coil voltage for different types of relays
- Measure the current consumption of the relay coil
- Measure the impedance of a coil
- Accurately measure time
    - Recording time stamps
    - Calculating time for tests
    - Repeating tests at equal intervals
- PC must automatically detect that the relay analyzer is connected

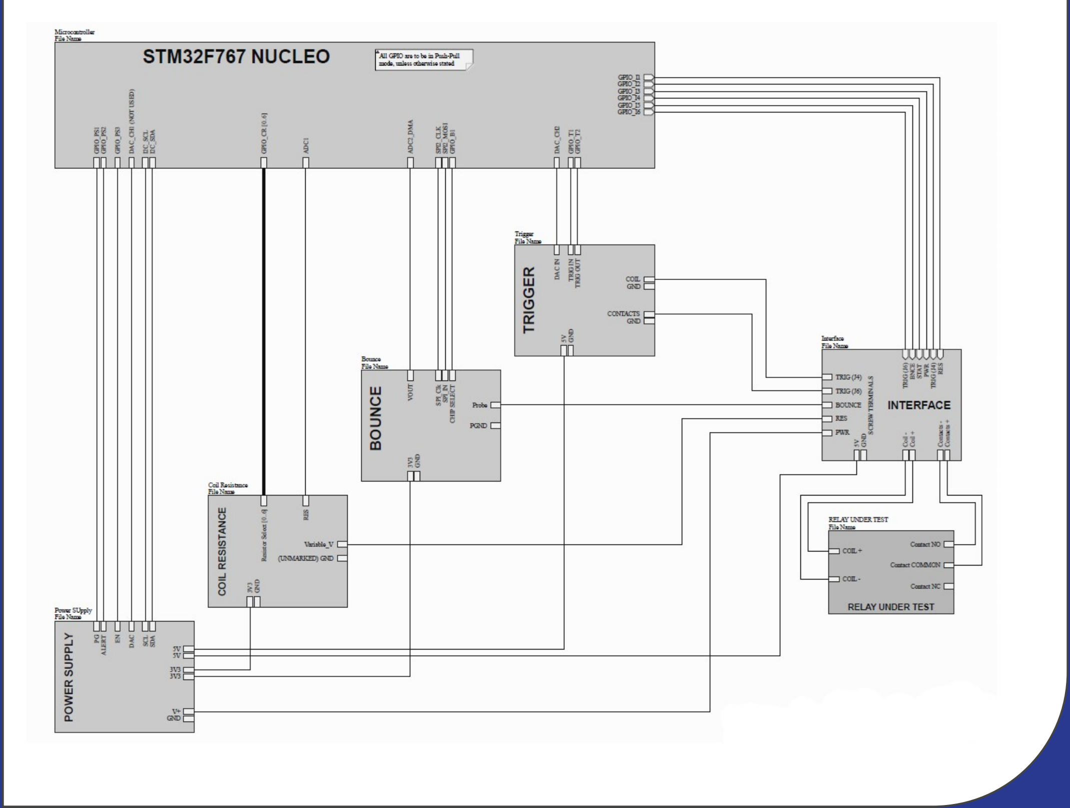
Project Requirements

- Perform tests at specified intervals
- Use an off-the-shelf power AC/DC supply
- Interface with a PC
  - Send collected data to a PC
  - Receive commands from a PC
  - Communicate quickly
- Archive all data to a database
- Work with electromechanical relays
  - Power
  - Signal
  - Automotive

System Design - Hardware



System Design - Connection Diagram

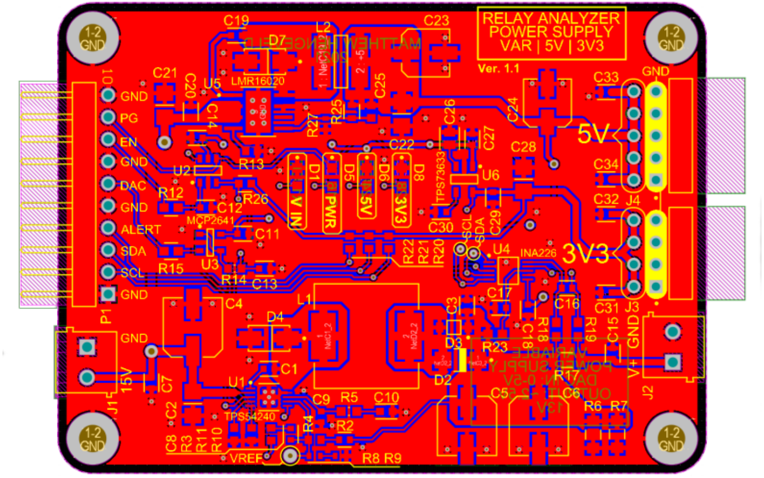


System Design - Microcontroller

- STM32F767ZI-Nucleo Microcontroller
- 216MHz system clock
- Large amount RAM
- Onboard DAC
- Fast ADC
- Real Time Clock (RTC)
- Onboard device USB port with magnetics
- Most peripheral pins on headers

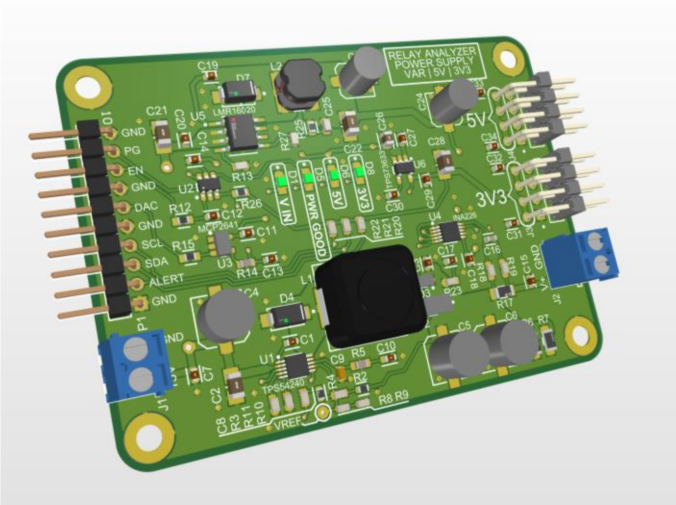
System Design - Hardware PCBs Overview

- PCBs designed in Altium Designer
- Manufactured by JLCPCB with solder stencils
- Parts from Digikey
- Reflow oven



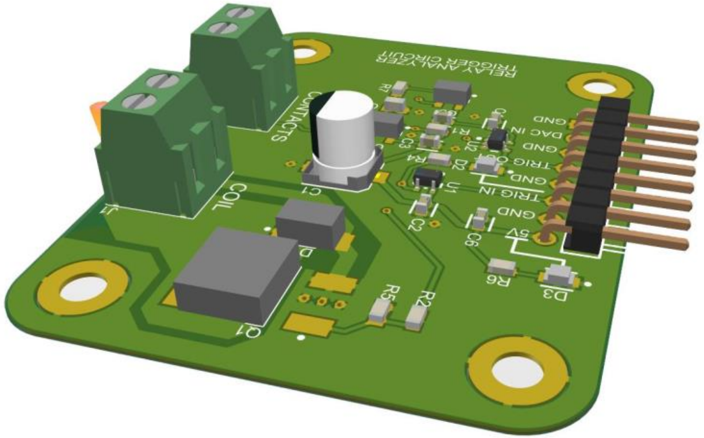
System Design - Programmable Power Supply

- Programmable output voltage
  - TPS54240 switching controller
  - ~2V to ~14.5V output up to 500mA supports 3V to 12V relays
  - DAC on microcontroller affects voltage on switching controllers Vsense pin
- 3V3 and 5V output
  - LMR16020 switching controller
  - Up to 1A
- Current & voltage monitor
  - INA226



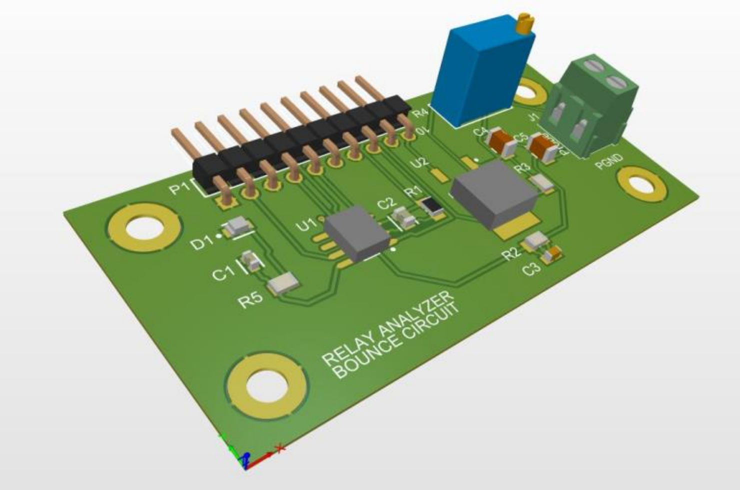
System Design - Trigger

- N-Channel MOSFET to switch relay coil
- Input and output Schmitt triggers
- Comparator on relay contacts

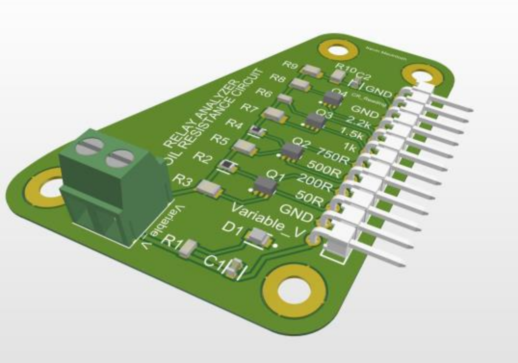


System Design - Bounce

- Analog front end to the contact bounce signal ADC
- MCP6S21 Programmable Gain Amplifier (PGA)
    - At least 2MHz bandwidth
    - 1x to 32x gain
    - SPI gain programming

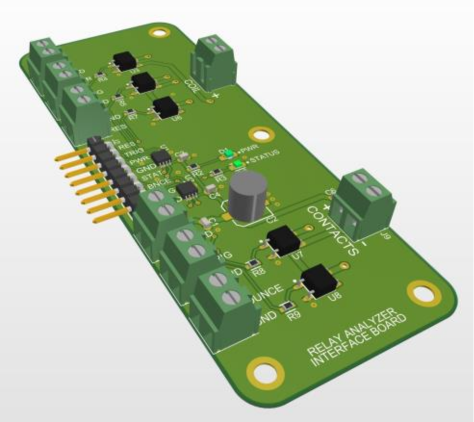


System Design - Resistance and Interface



Coil Resistance circuit

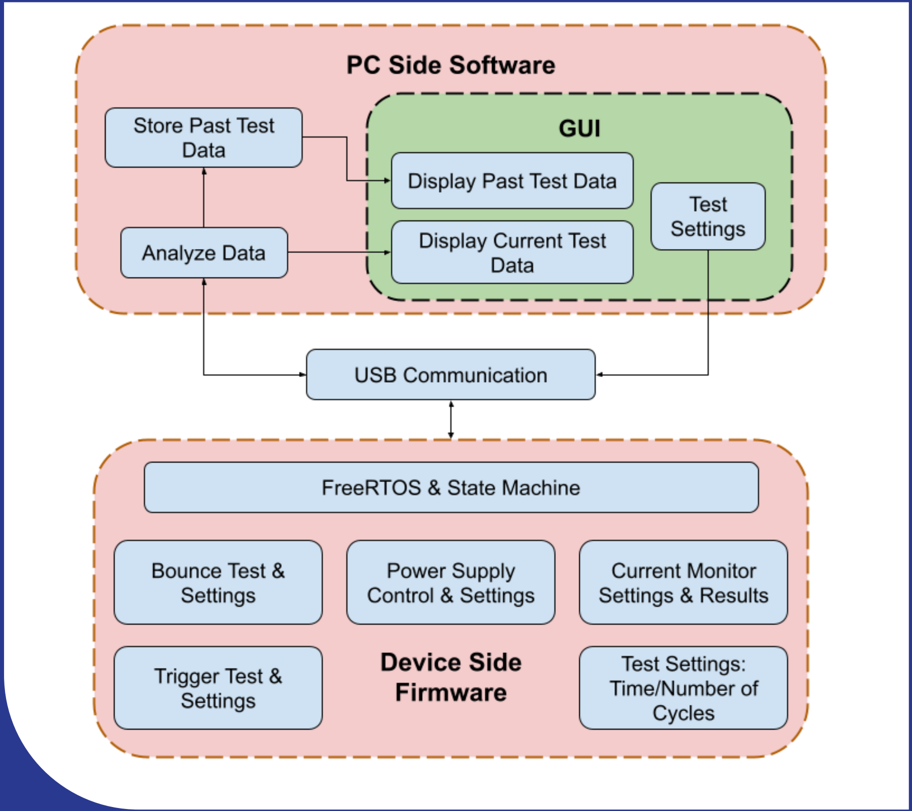
- Compares reference resistors to coil resistance
- Voltage read by ADC is used to calculate resistance



Relay Interface

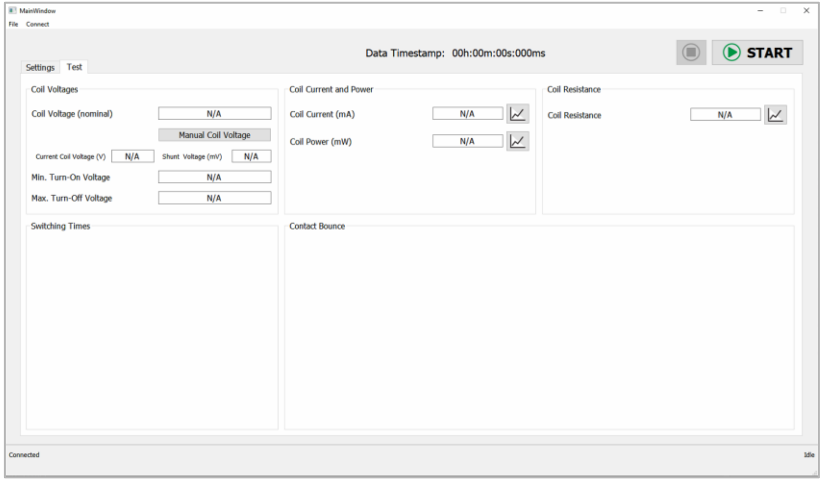
- Simplifies connection of all circuits in system
- Switches connection to relay for different tests using SSRs

System Design - Software



System Design - Software

- Graphical User Interface
- Automatically connects to Relay Analyzer
- Allows user to change Relay Analyzer settings
- Start and stop test
- Displays current cycle test results
- Displays graphs of past cycle test results
- Qt framework
- libusb-win32



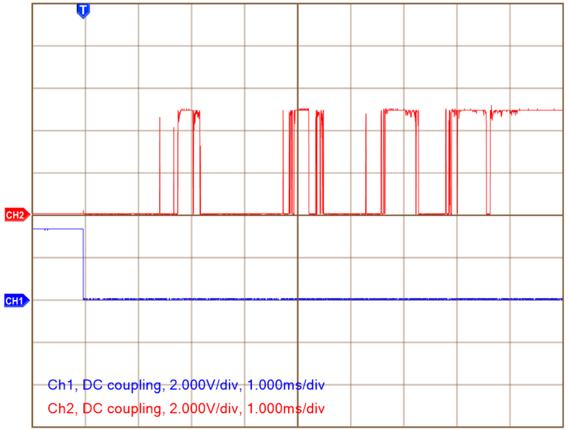
System Design - Firmware

- STM32CubeMX for system and peripheral setup
- STM32 Hardware Abstraction Layer (HAL)
- Custom USB full-speed device code
- FreeRTOS real-time operating system
  - Scheduled tasks
  - Required for USB function

Test Methodology

Individual Circuits

- Tests run prior to manufacturing with simulation software
- Visual inspection after soldering
- Electrical test to compare with simulation results
- Test without microcontroller using test equipment
- Test with microcontroller and developed firmware



Test Methodology

Firmware

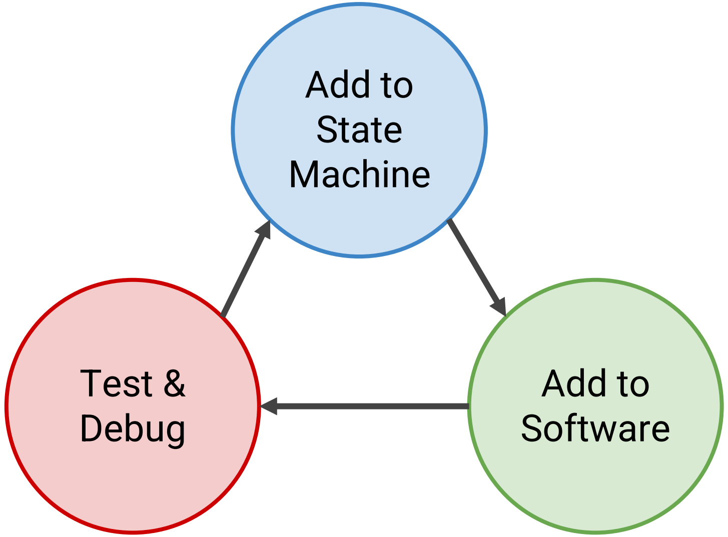
- Individual peripheral code tested separately and with generic hardware
- Peripheral code combined & adapted for each PCB
- USB code tested separately as part of Adv. Elective course

```
// get Bus Voltage
i2cBuf[0] = 0x02;
HAL_I2C_Master_Transmit(&hi2c1, INA226_ADDRESS_READ, i2cBuf, 3,
INA226_I2C_TO);
HAL_I2C_Master_Receive(&hi2c1, INA226_ADDRESS_READ, &i2cBuf[1], 2,
INA226_I2C_TO);

i2cresults[0] = (uint16_t) i2cBuf[1] << 8 | i2cBuf[2];

// get shunt Voltage
i2cBuf[0] = 0x01;
HAL_I2C_Master_Transmit(&hi2c1, INA226_ADDRESS_READ, i2cBuf, 3,
INA226_I2C_TO);
HAL_I2C_Master_Receive(&hi2c1, INA226_ADDRESS_READ, &i2cBuf[1], 2,
INA226_I2C_TO);

i2cresults[1] = (uint16_t) i2cBuf[1] << 8 | i2cBuf[2];

// get Bus Power
i2cBuf[0] = 0x03;
HAL_I2C_Master_Transmit(&hi2c1, INA226_ADDRESS_READ, i2cBuf, 3,
INA226_I2C_TO);
HAL_I2C_Master_Receive(&hi2c1, INA226_ADDRESS_READ, &i2cBuf[1], 2,
INA226_I2C_TO);

i2cresults[3] = (uint16_t) i2cBuf[1] << 8 | i2cBuf[2];

// get Bus Current
i2cBuf[0] = 0x04;
HAL_I2C_Master_Transmit(&hi2c1, INA226_ADDRESS_READ, i2cBuf, 3,
INA226_I2C_TO);
HAL_I2C_Master_Receive(&hi2c1, INA226_ADDRESS_READ, &i2cBuf[1], 2,
INA226_I2C_TO);

i2cresults[2] = (uint16_t) i2cBuf[1] << 8 | i2cBuf[2];
```

Test Methodology

System

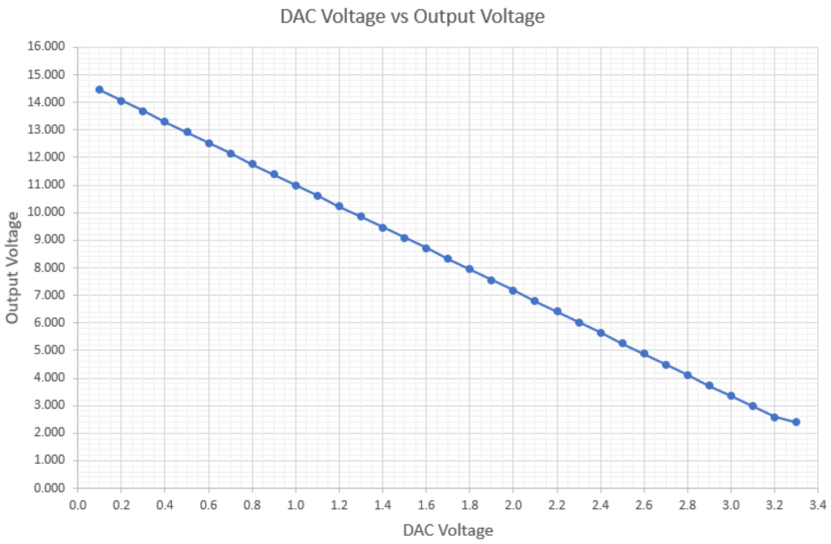
- Microcontroller state machine continually built upon until all individual components were added
- Software was developed in parallel
- Continuous testing



Results and Analysis - Key Components

Programmable Power Supply

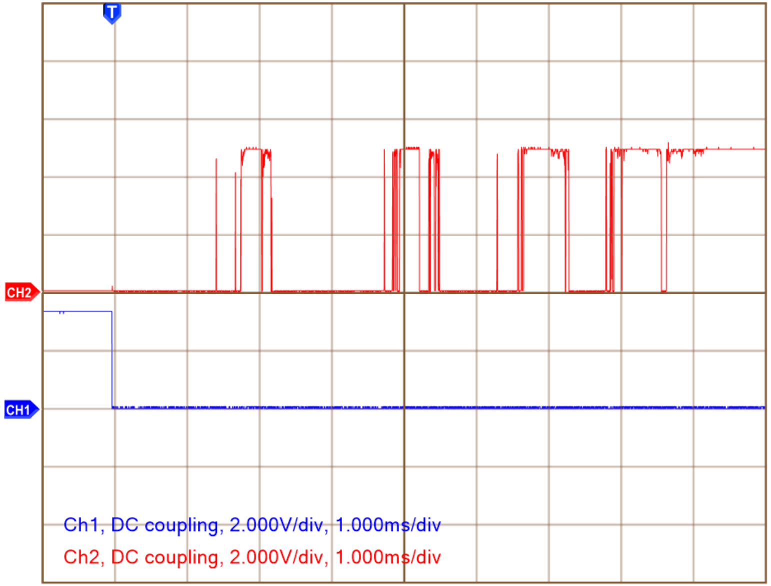
- 2.1V to 14.5V output voltage
- Linear relationship between DAC voltage and output voltage



Results and Analysis - Key Components

Trigger Circuit

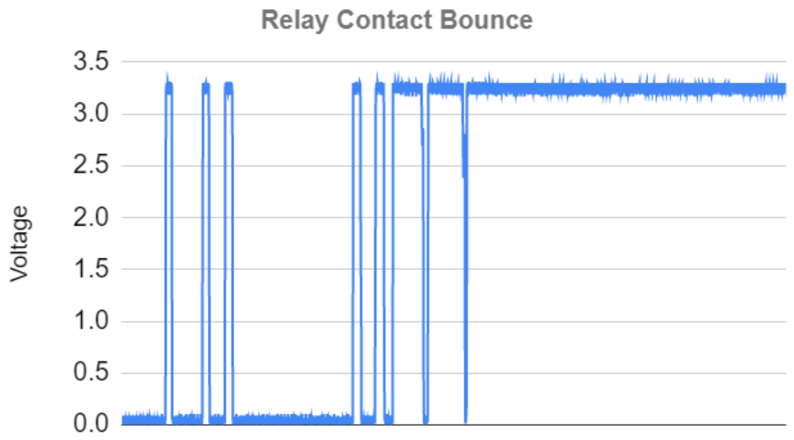
- Fast switching & sharp edges
- Output shows accurate relay contact bounce characteristic and switching times



Results and Analysis - Key Components

Bounce Circuit & ADC

- ADC accurately captures bounce waveform
  - Sufficient samples per second to capture all features of bounce waveform
- Relay contact signal scaled to ADC full-scale voltage for better resolution



Bounce data output from microcontroller ADC

Demonstration

Going Forward...

- Implement full database capabilities
- Design single PCB that combines all PCBs & microcontroller
- Design proper casing for device
    - Hold PCBs
    - Proper connections for device under test
- Widen range of compatible relays
- More user-friendly and feature-rich GUI
- Measure more than one relay at a time
    - Compare multiple relays characteristics

Thank You