**LabVIEW 101 – Weeks 8, 9**

Before class:

1. Download updates from <https://github.com/rizett/LabVIEW-101>

Required equipment:

Computer with LV and NI-MAX installed

3x Fans

3x DC power supplies

3x DC relays

1x Solenoid valve

1x Thermocouple

2x Paddlewheel

1x Optics valve

Topics:

Sending and acquiring analog signals via the NI-DAQ

* Week 8: Using the NI-DAQ to control power output to a device (e.g. fan, pump, valve)
* Week 9: Using the NI-DAQ to read voltage from a device (e.g. thermocouple, paddlewheel)

Think about your projects!

**Week 8: Sending signals via the NI-DAQ**

1. Using NI-Max:
   1. Wire output from NI-DAQ to Relay
   2. Draw a diagram!
   3. Use NI-MAX to control the Relay: select device > test panels > analog output
   4. Wire in a fan
2. Repeat using LabVIEW
   1. Setting up NI-DAQ VI: Block diagram > Measurement I/O > NI-DAQmx > DAQ Assistant
      1. Acquire signals > analog input > voltage (all other options are for NI-specific devices) > select physical channel
   2. Input information
      1. Data (0 or 5 V): merge signals if controlling 2 channels (Express > Sig Manip > merge)
      2. Device name (IO control on front panel > I/O > DAQmx Name Controls > DAQmx Device Name) (self-populates)
   3. Create a simple VI: test with fan

Exercise:

With a partner: create a **subVI** to control when to turn a device on / off. Use either a fan or valve.

Allow the device to be controlled manually (i.e. using a user-defined Boolean switch) AND automatically (i.e. when a signal exceeds or falls below a certain threshold). Use a case structure to separate these two sets of actions and a front panel control to select which “mode” to use. In the automatic mode, enable a manual over-ride so that the device can be turned off by a user, or when the threshold is exceeded (requires an OR statement somewhere…).

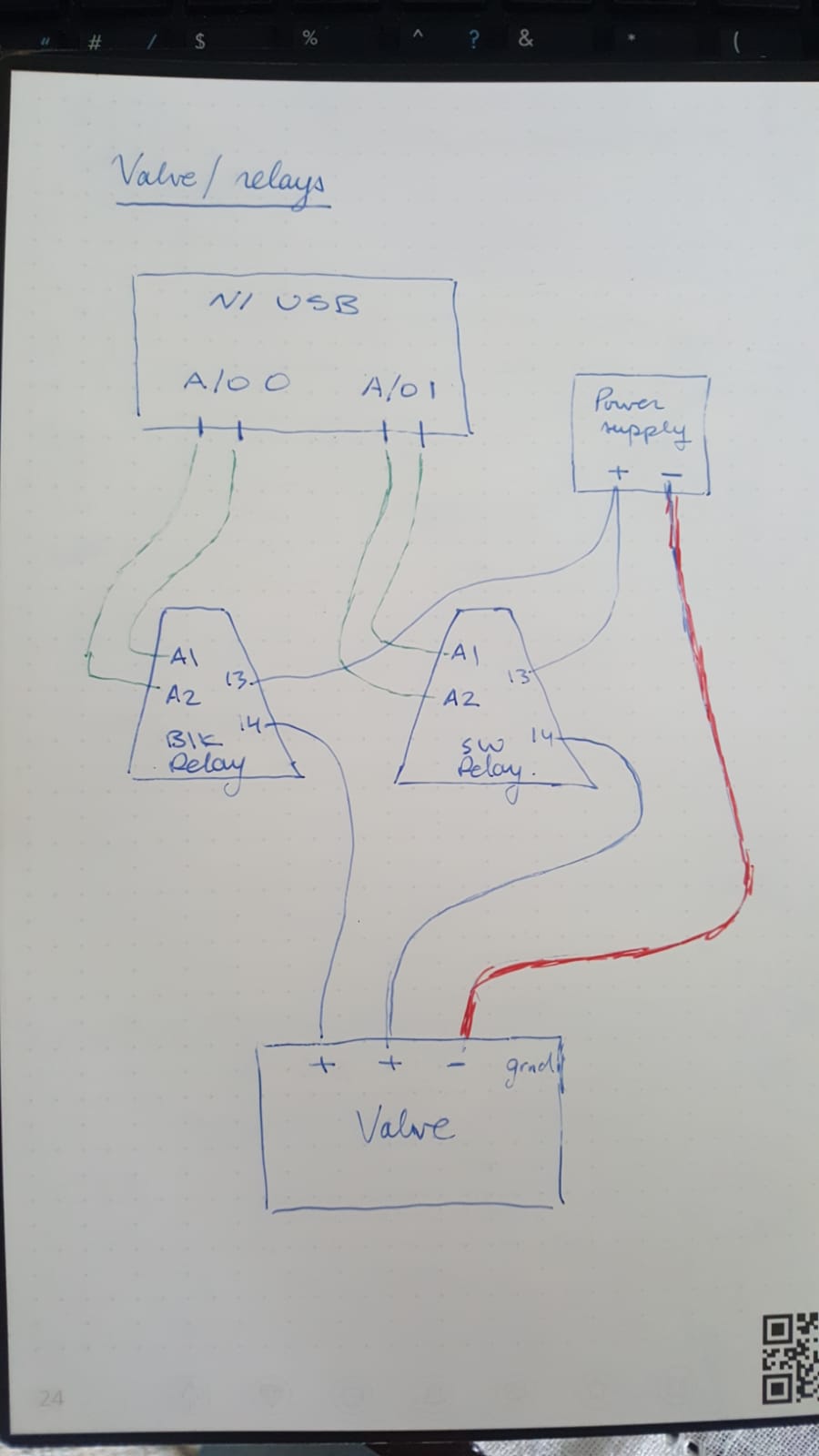
Include a Boolean LED indicator to show when the device is on.

Think about appropriate terminals to wire in / out of the subVI so that it can be integrated into a larger VI program, and used generically to control different devices.

Save to the LabVIEW 101 library.

**Week 9: Review AO**

1. Review wiring for AO
2. Optics valve (during exercises)



**Week 9: Acquiring signals with NI-DAQ**

*One group use thermocouple; one use paddlewheel*

1. Review wiring / electrical specifications for each device: power requirements, signals etc.
   1. Connect wires to power supply and NI-DAQ
   2. Draw a diagram
2. Using NI-Max:
   1. Use NI-MAX to acquire a signal: select device > test panels > analog input
3. Repeat using LabVIEW
   1. DAQ Assistant: Block diagram > Measurement I/O > NI-DAQmx > DAQ Assistant
      1. Acquire signals > analog input > voltage (all other options are for NI-specific devices) > select physical channel
      2. \*Consider the sampling mode! N Samples, 1 Sample, Continuous…
   2. Input information
      1. Device name (IO control on front panel > I/O > DAQmx Name Controls > DAQmx Device Name)
      2. Number of samples to acquire
      3. Sampling rate (in Hz)
   3. Output information
      1. Data!
   4. Create a simple VI to read the data from your device: plot the output vs time/Julian Day
      1. Play with different sampling options (N samples, 1 sample etc.)

Exercises:

Switch devices (thermocouple vs paddlewheel) with the other group.

1. Build a subVI that will read the data from the device. It’s up to you if you would also like to plot the data in the subVI (*for A+ marks, I’d add the* *graph … it can be helpful for troubleshooting / testing a device in a stand-alone subVI*).

🡪 Consider the NI-DAQ sampling mode (e.g. continuous, 1-sample, N-samples etc.)

🡪 Make sure that you include some appropriate front panel controls and indicators to show a) what voltage is being read from the NI-DAQ, b) a user-defined sampling rate, c) the device name, d) the physical channel number (?)

🡪 Wire these inputs / outputs through the subVI icon

🡪 Save the subVI to the LabVIEW 101 library: read-thermocouple\_subVI or read-paddlewheel\_subVI

1. Create a new VI that you will use to calibrate the device. *For the thermocouple, temperature can be related to the output voltage with a simple linear equation (T = m\*V + b). For the paddlewheel, flow rate is related to the number of step changes by a “K-factor” (# steps / L)*

🡪 Insert the subVI from above into a while loop

🡪 Add a data saving mechanism

🡪 Use a boolean to will allow you to manually select when to write data to a file (e.g. only write data when you press OK)

🡪 Write the following to your file:

* Column 1: output voltage (thermocouple) or number of steps (paddlewheel)
* Column 2: real temperature (determined with thermometer) or flow rate (determined with graduated cylinder)

🡪 test the VI! (don’t need to fully calibrate the device … though you may want to down the road)

🡪 Save the VI to the LabVIEW 101 library: calibrate-thermocouple or calibrate-paddlewheel

🡪 Share your subVI / VIs with the other group

1. Create a new VI that reads temperature and flow rate from the NI-DAQ and information from the provided calibration file thermocouple\_calibration.csv (LabVIEW 101 > examples & functions > examples) (recall the subVI you created in week 6!)

🡪 Convert the thermocouple voltage to a temperature value using the calibration information

🡪 Convert the paddlewheel # of steps to a flow rate using a user-defined (front panel control) K-factor

🡪 Wire in a relay to the NI-DAQ AO, and add your subVI from last week. Use either the temperature or flow rate to provide feedback to turn off/on the relay when the data exceeds a user-defined threshold.

🡪 Save the VI to the LabVIEW 101 library: week9\_read-T-flow\_write\_NI-DAQ.vi

Make sure that both partners have copies of the exercises.