**LabVIEW 101 – Weeks 10, 11**

Before class:

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

* [https://sourceforge.net/projects/realterm/files/latest/](https://sourceforge.net/projects/realterm/files/latest/download)
* NOTE: going to this link will start download automatically.

Required equipment:

Computer with LV and RealTerm installed

3(6)x Valco Valves + cables, supplies etc.

3x 24VDC power supplies

1x Keyspan

3(6)x USB-to-Serial

Topics:

Sending and reading data via serial

* Week 10:

Interfacing with serial devices in Real Term

Setting up serial communication in LV

Using Valco Valves to control inflow/outflow sample

* Week 11:

2-way communication between serial device and computer (Optode/GTD; Winkler Pump/Meter)

Think about your projects!

**Week 10: Sending serial commands/information**

1. Setup Valco valves – may need wiring to power supplies
2. Using RealTerm
   1. Check com-port using converter and keyspan (Device Management)
   2. RealTerm settings
   3. Sending commands
3. Repeat using LabVIEW
   1. Comp port: (FP): Modern > IO > VISA Resource Name
   2. NI-VISA: (BD) Instrument IO > VISA > Serial
      1. **Configure Port**: set up port settings
         1. Inputs: timeout, Port (VISA resource), Baud, Bits, Parity, Stop bits (optional: termination character)
      2. **Flush buffer**: clear read/write buffer
         1. Inputs: VISA resource
      3. **Write**: Write to serial
         1. Inputs: VISA resource; Command **with termination character**
      4. **Read**: Read from serial
         1. Inputs: VISA resource
         2. String out
      5. Note: VISA resource is passed from one VI to the next
   3. Create a simple VI to send a manual command to the Valco Valve when you push a button. Return the valve’s position as a numeric indicator.

Exercise:

1. Create a subVI that controls the Valco Valve position and returns the valve position.

The subVI should receive a **cluster** of information as input (i.e. wired input) that contains: the baud rate, comp port / VISA resource, data bits, stop bits and parity. Also wire in the desired valve position.

The subVI will then send the appropriate command to the valve to change the position. In the same subVI use the “CP” command to retrieve the valve’s current position. Return that value as a **numeric indicator** (you’ll have to parse out the numeric position from a string) output from the subVI.

Save the subVI to your LabVIEW101 library as week10\_valco\_valve\_subVI

1. Create a new VI that runs the subVI above.

Create a text ring (or similar) to define 2 cases: a manual mode and an automatic mode.

In the manual mode, a user will be able to change the valve position on-demand.

In the automatic mode, the VI will run an automated calibration sequence where the valve position will change automatically at a user defined interval.

Plot the valve position versus time.

Save the VI to your LabVIEW101 library as week10\_valve\_position\_VI