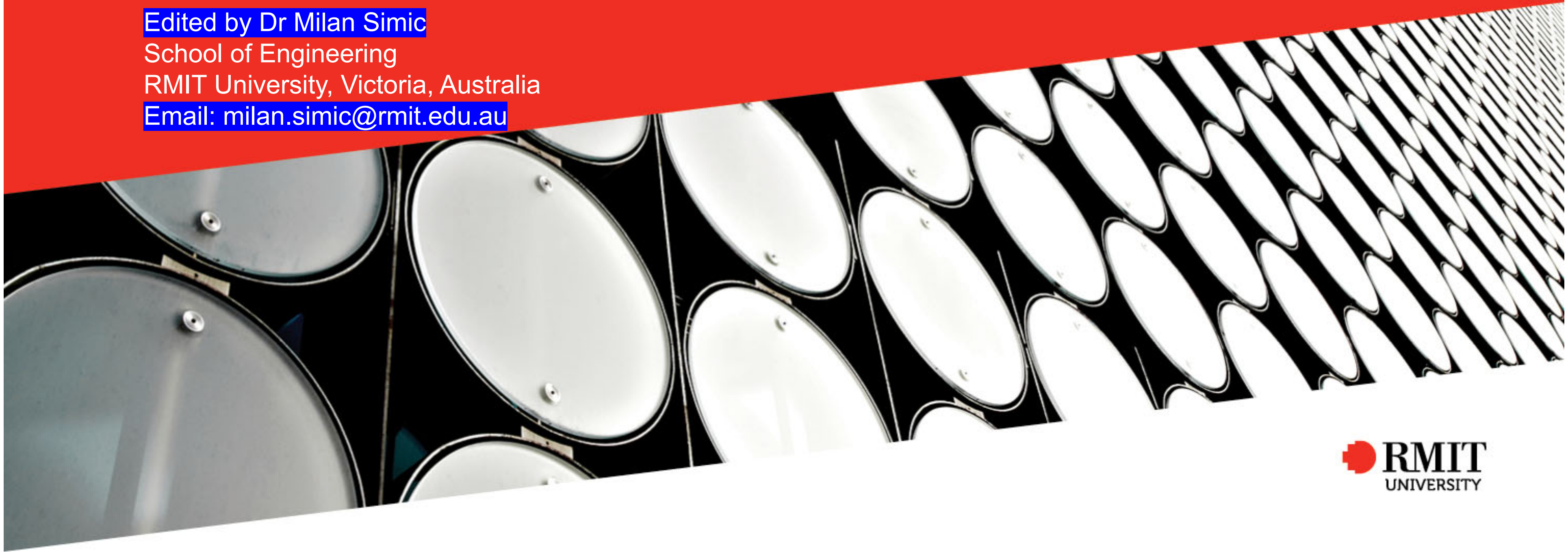
Week 5 – Sensors II

Advanced Mechatronics System Design – MANU2451

Dr Chow Yin LAI
Edited by Dr Milan Simic
School of Engineering
RMIT University, Victoria, Australia
Email: milan.simic@rmit.edu.au



RMIT UNIVERSITY

New Teaching Schedule

Week		Class Activity Before	Lecture	Class Activity During or After
1			Introduction to the Course / Introduction to LabVIEW	LabVIEW Programming
2			Introduction to LabVIEW / Data Acquisition	LabVIEW Programming
3			Gripper / Introduction to Solidworks / Safety	Gripper Design
4			Sensors I	myRIO Programming for Sensor Signal Reading / Gripper Design
5			Sensors II	myRIO Programming for Sensor Signal Reading
6			Actuators I	LabVIEW Tutorial
7		LabVIEW Assessment.	DC Motors I	Matlab Simulink Simulation
8		Design report submission	DC Motors II	Matlab Simulink Simulation / myRIO Programming for Control
9			Actuators II	Matlab Simulink Simulation Gripper CAD
10			Modeling and System Identification	Matlab Simulink Simulation / Gripper simulation testing
11			Artificial Intelligence I	Matlab Simulation / Finalize Gripper
12		Gripper Simulation / Submission of Report	Artificial Intelligent II	Revision

Assessment 2020

- LabView assessment (Individual online Quiz) on week 7 (22/04) – **20**%

Design assessment (Team) on Tuesday of week 8 (29/04) – **10**%

  - A brief written Report, with figures of the gripper assembly, and explanation of the components.
  - Also, explain choice of design, calculation etc.

- Submit Gripper Simulation Program and Report (Team) on week 12 (27/05) **10%** and **40%** respectively.
  - Report template is given in the course folder.
- Final exam (Individual) – **20**%
  - Online Quiz .
  - Will be on Wednesday of Week 14.

Peer Assessment

Note that the team assignments, i.e.

- Design assessment (10%)
- Gripper demonstration and final report (50%)

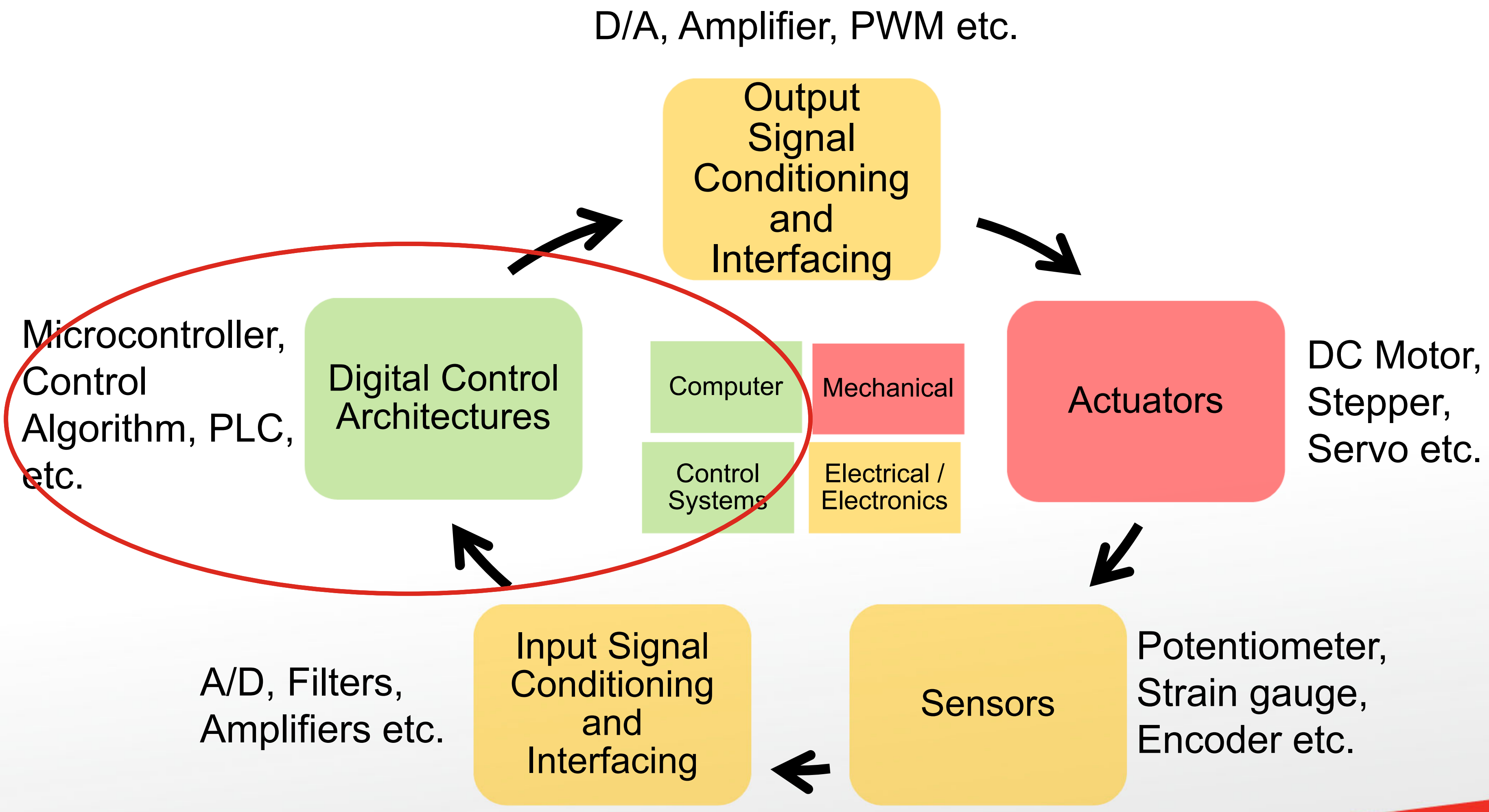
will have a peer assessment component.

How it works:

- You will be given a group score after the initial marking.
- In Week 12 or 13, I will send out a questionnaire where you will be rated by other team members based on your participation and contribution.
- The final individual score will be then :
    - (group score) x (average rating by other team members).

RMIT UNIVERSITY

Mechatronics System Components



Project Options – Control Design

In a myRIO project, the program can run from three different “locations”:

1. **PC (host):**
    1. Suggested: For non-time-critical tasks such as complex UI to visualize data.
2. **Real-time processor:**
    1. Suggested: For time-critical tasks which needs to run in real-time, i.e. cannot be interrupted by lower priority tasks (anti-virus, UI etc.)
    2. Also for data logging, file management.
3. FPGA:
    - Suggested: For tasks which are time-critical and has to be very fast, e.g. Accessing IO, PWM, reading encoder pulses, emergency stop.

In this online course, we will develop simulation program to run on

- PC (Option 1) or,
- myRIO real, or simulated (Option 2)
- Programming environment could be LabVIEW, MATLAB/Simulink, C/C++, JAVA, Pyton… // Objective is the *Development of Programming Mindset*
- CAD design will not be printed, just submitted

Notes on programming in FPGA will also be given but these are only for self-study. (Please refer to attachment).

RMIT UNIVERSITY

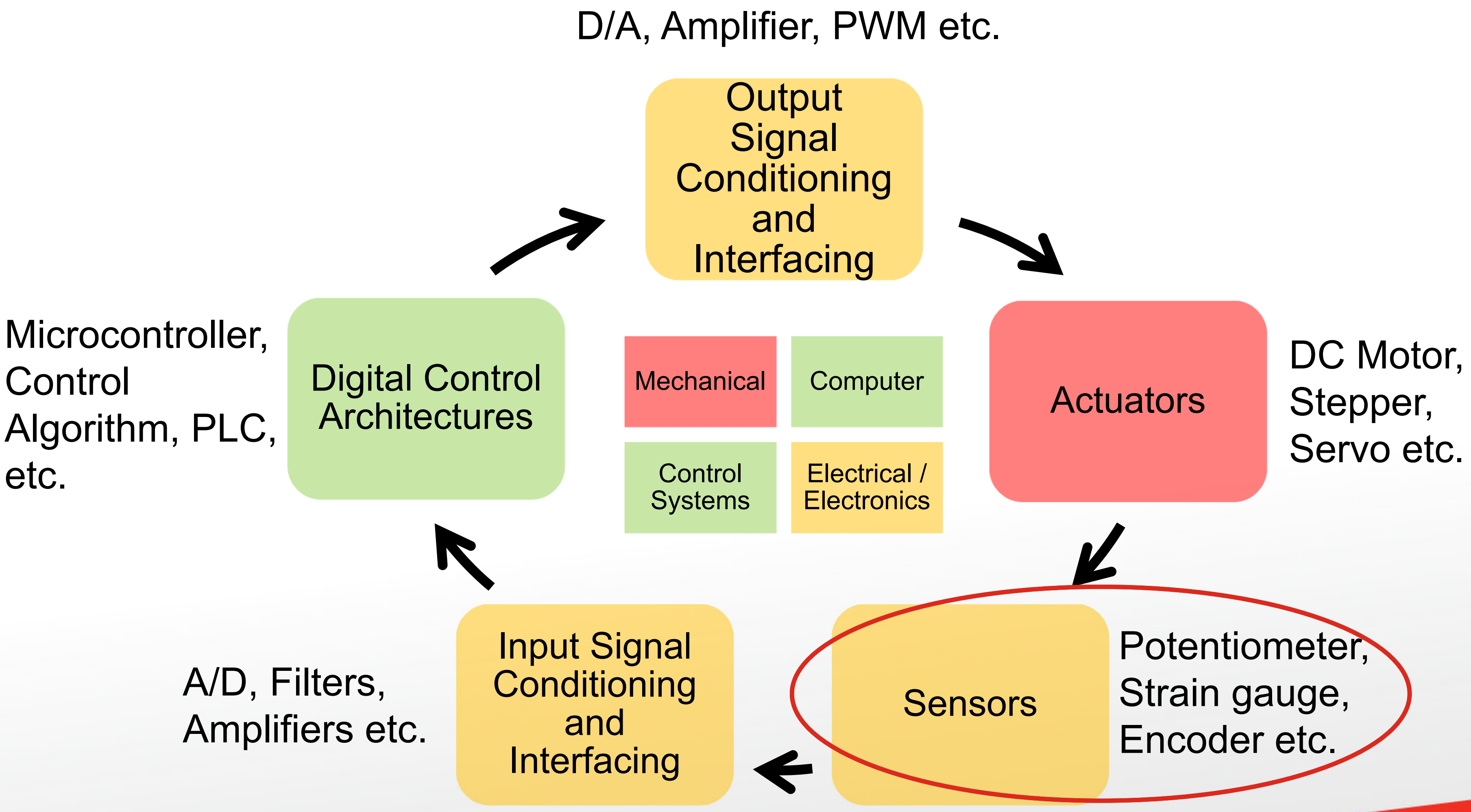
Online Collaboration

- Collaboration
- Discussions
- Collaborate Ultra
- Microsoft Teams
- Skype
- Viber
- FaceTime
- Email
- …
- Teams and the team members / *please send me email (just 11)*

Sensors and Actuators

- Sensors and actuators are used to transform
  - signals and energy carried by a physical quantity of one kind
  - into signals and energy carried by a physical quantity of another kind.
- There are many different types of sensors and actuators in use.

Mechatronics System Components



Mechatronics System Components

Sensors:
Encoder at
each joint

Input signal interfacing

Robot controller:
- Generate desired motion trajectory
- Calculate current end-effector position based on angular position (kinematics)
- Calculate desired angular position for desired end-effector position and trajectory (inverse kinematics)
- Control algorithm
- Safety, collision detection etc.

Output signal interfacing

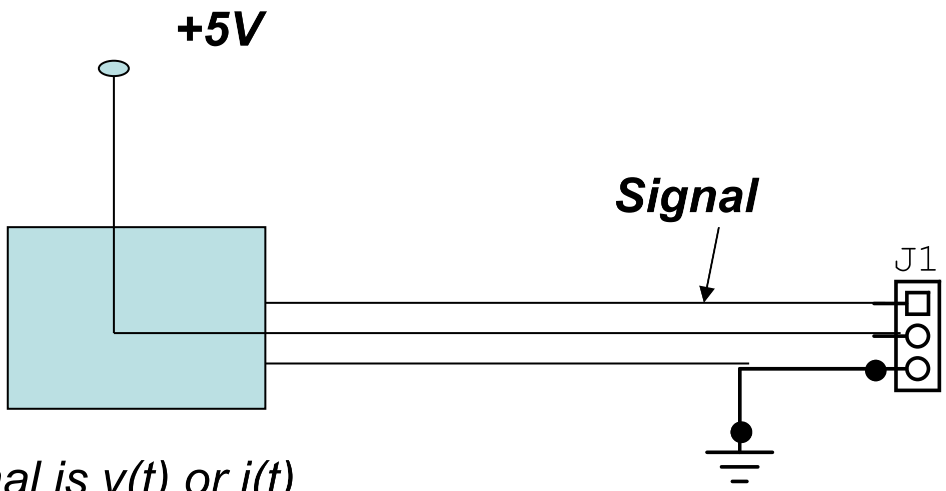
Actuators:
Geared motor
at each joint

Industrial Robots
https://commons.wikimedia.org/wiki/File:Float_Glass_Unloading.jpg

Variety of Sensors

- Transducers
    - Motion
    - Temperature
    - Force
    - Stress
    - Flow
- Position sensing
- Laser Range Sensors
- Smart Sensors

Most Common Sensor Connection



Signal is v(t) or i(t)

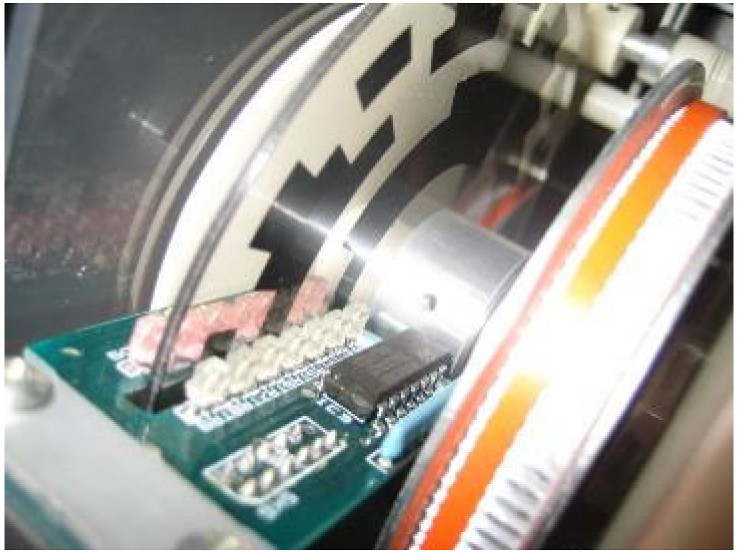
v(t) = function (measured quantity)

Or

i(t) = function (measured quantity)

Could be analogue or digital

Encoders,
Gray Code Instead of Binary
0,1
00, 01, 11, 10   WHY?



Encoder

Content

- Position Measurement
  - Measure using myRIO
- Speed Measurement
- Vibration and Acceleration Measurement
- Stress and Strain Measurement
  - Measure using myRIO
- Saving (Sensor) Data using myRIO
- Attachment: Writing FPGA Codes

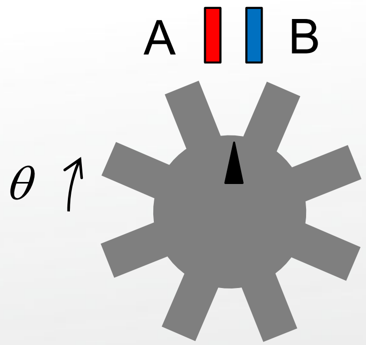
Content

- Position Measurement
    - Measure using myRIO
- Speed Measurement
- Vibration and Acceleration Measurement
- Stress and Strain Measurement
    - Measure using myRIO
- Saving (Sensor) Data using myRIO
- Attachment: Writing FPGA Codes

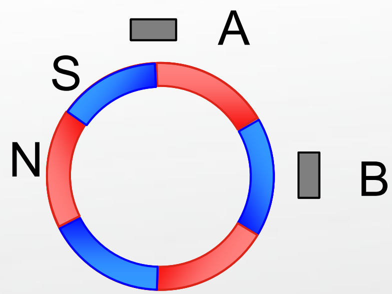
Position Sensors (4)

- Encoder
  - Converts motion into a sequence of digital pulses.
  - By counting bits, pulses can be converted to relative or absolute position measurements.
  - Can be linear or rotary.
  - Usage example:
    - Track the position of motor shaft in permanent magnet brushless motors.
    - Makes use of proximity sensor for the digital pulses.

Rotary Optical Encoder
https://commons.wikimedia.org/wiki/File:Encoder.jpg

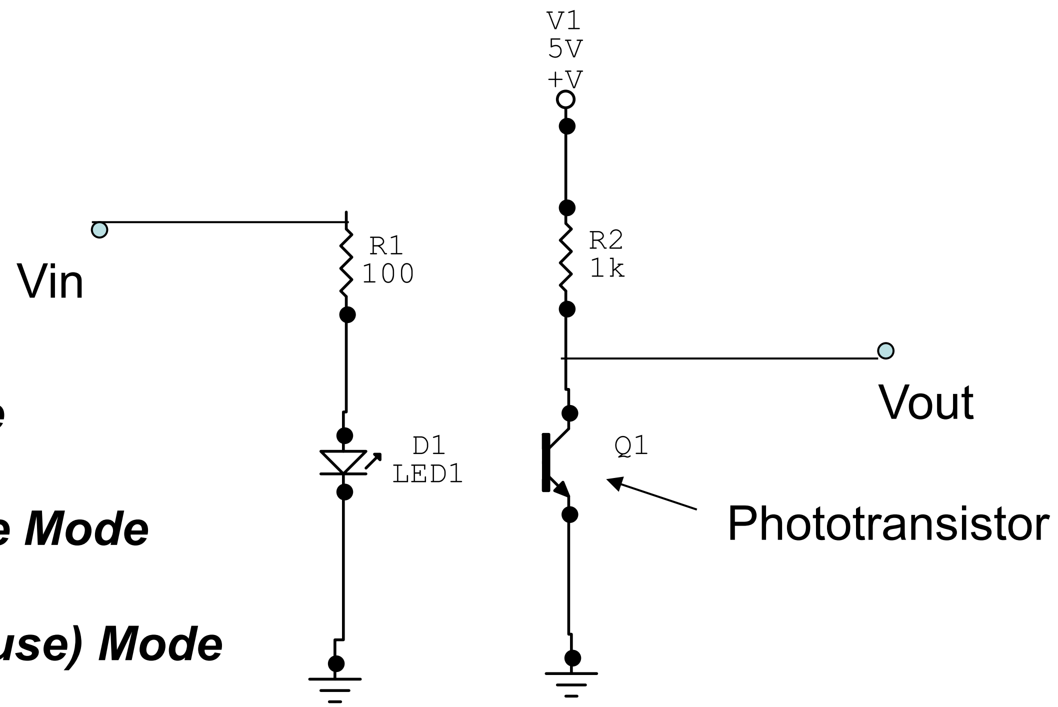


Optical Encoder



Hall Effect Encoder

Photo emitter-detector pair for Proximity Sensing



Opposed Mode

Retro-reflective Mode

Proximity (Diffuse) Mode

Distance Measuring – Sharp GP2D

Fig.1 Internal Block Diagram

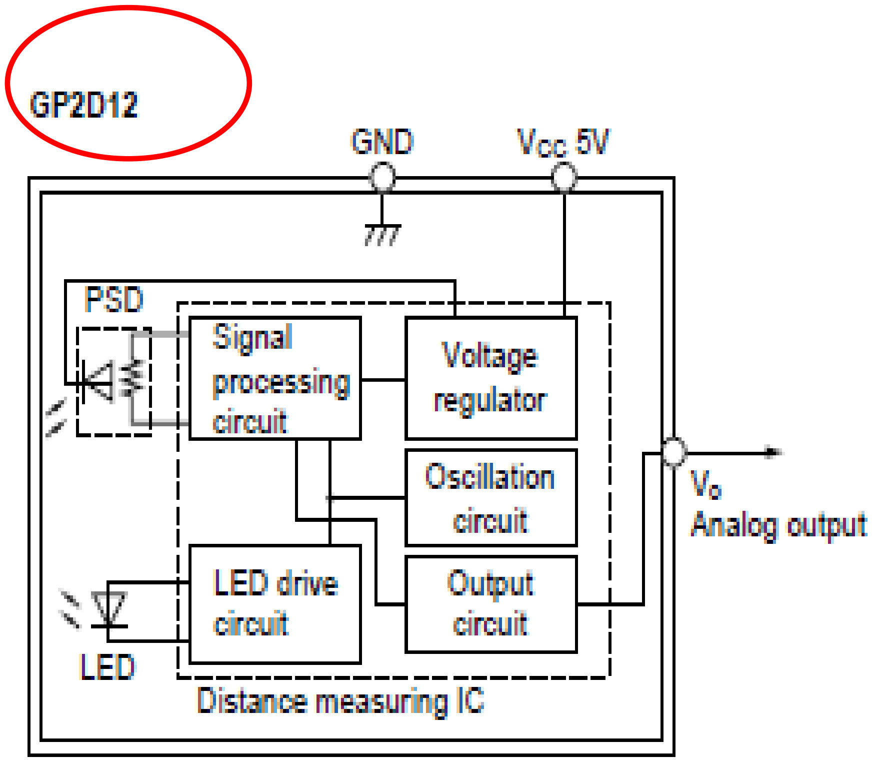


Fig.2 Internal Block Diagram

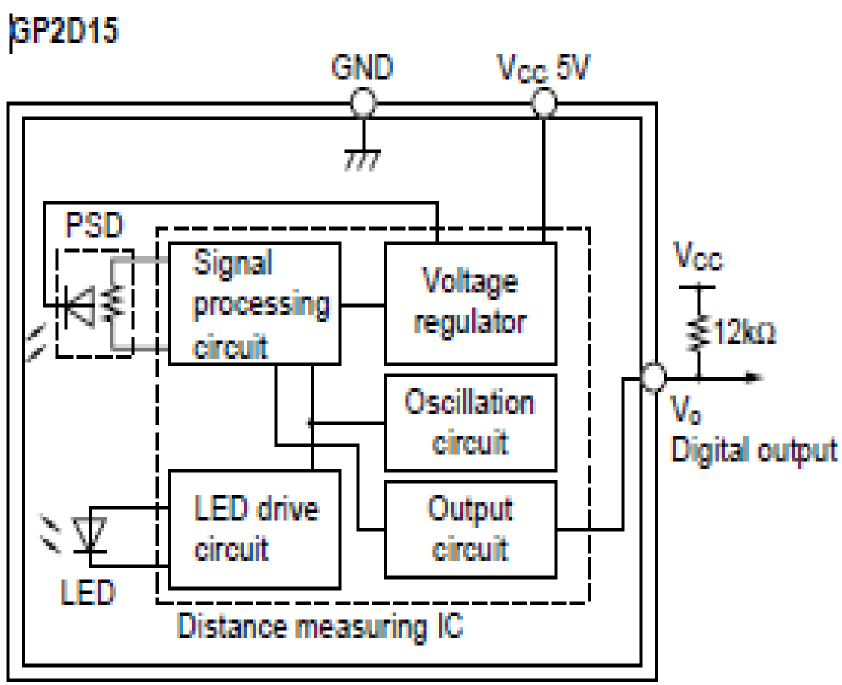
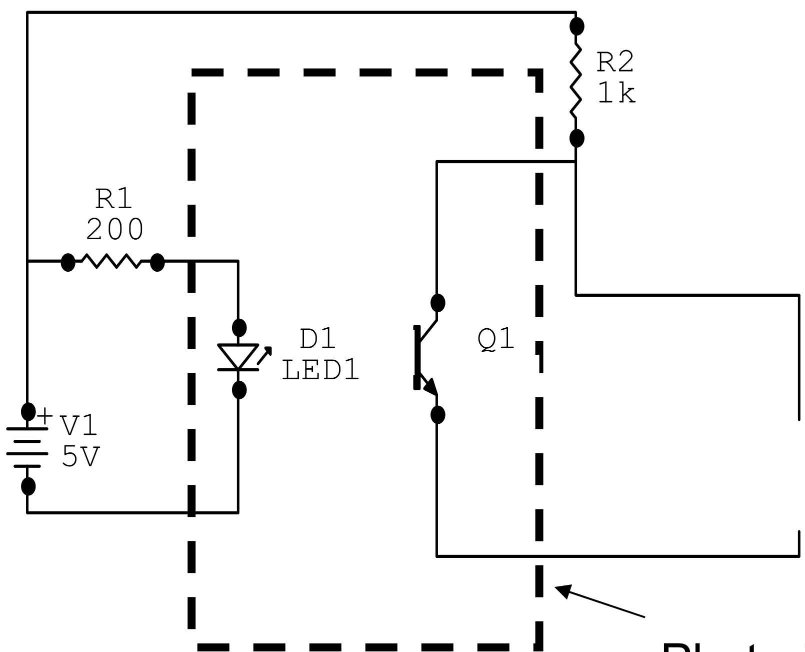


Photo-Sensing



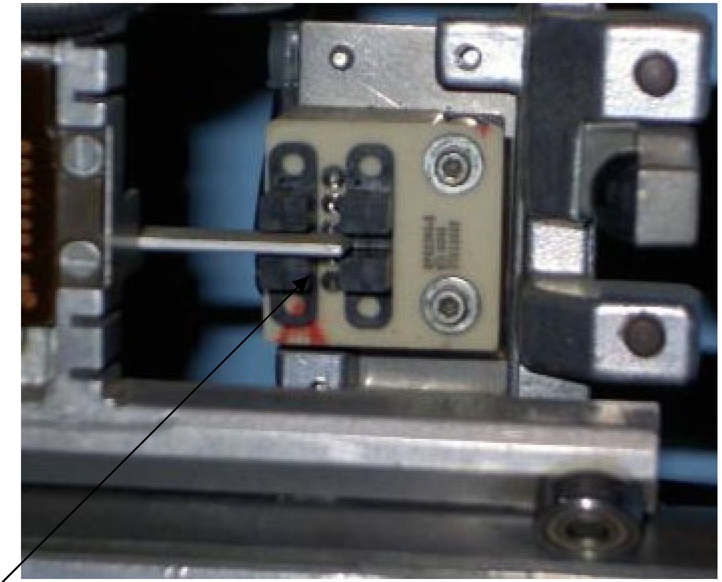
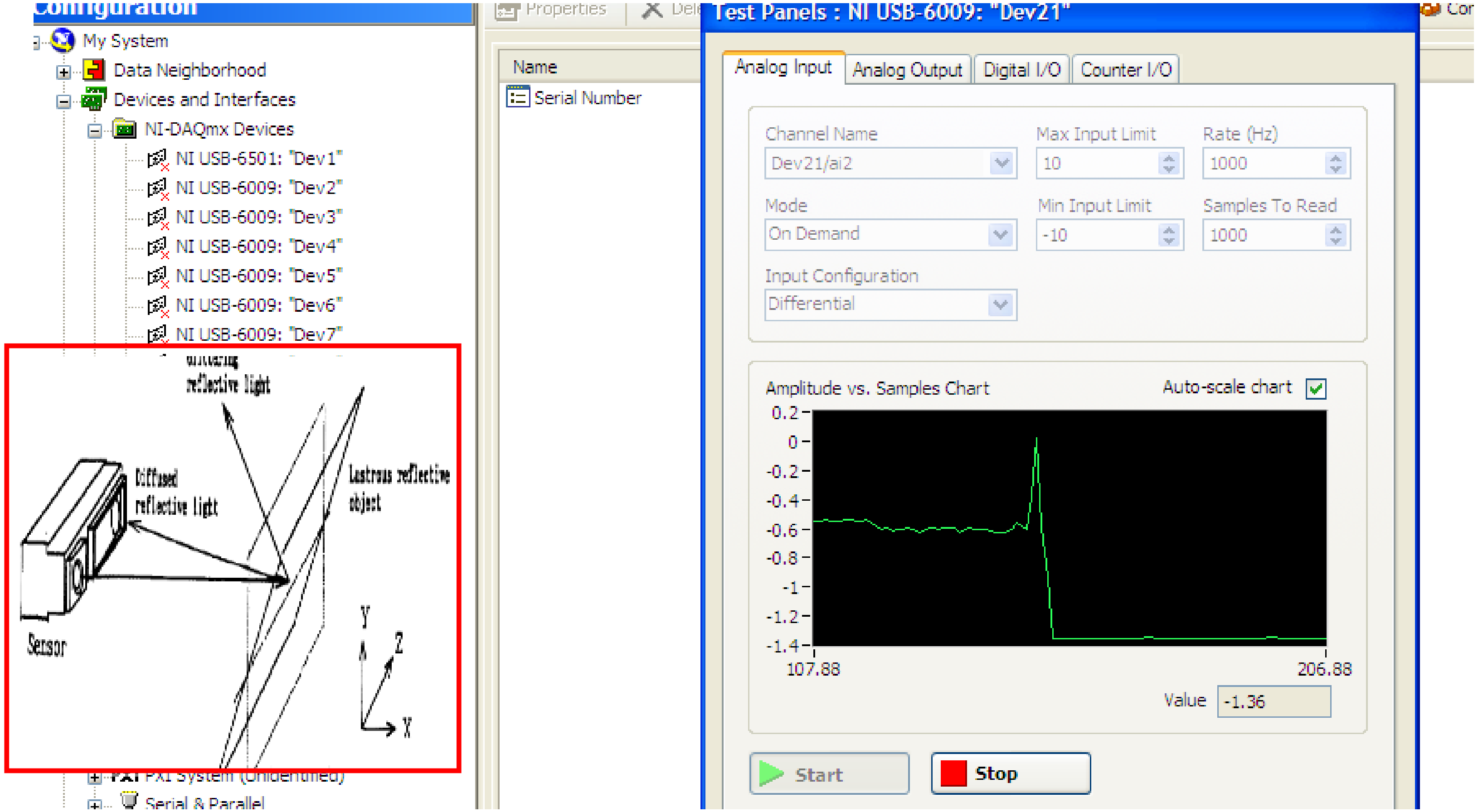


Photo-interrupter in a hard disc application

00, 10, 11 forward, back 11, 10, 00

Sensor Testing With MAX (Measurement & Testing Explorer)

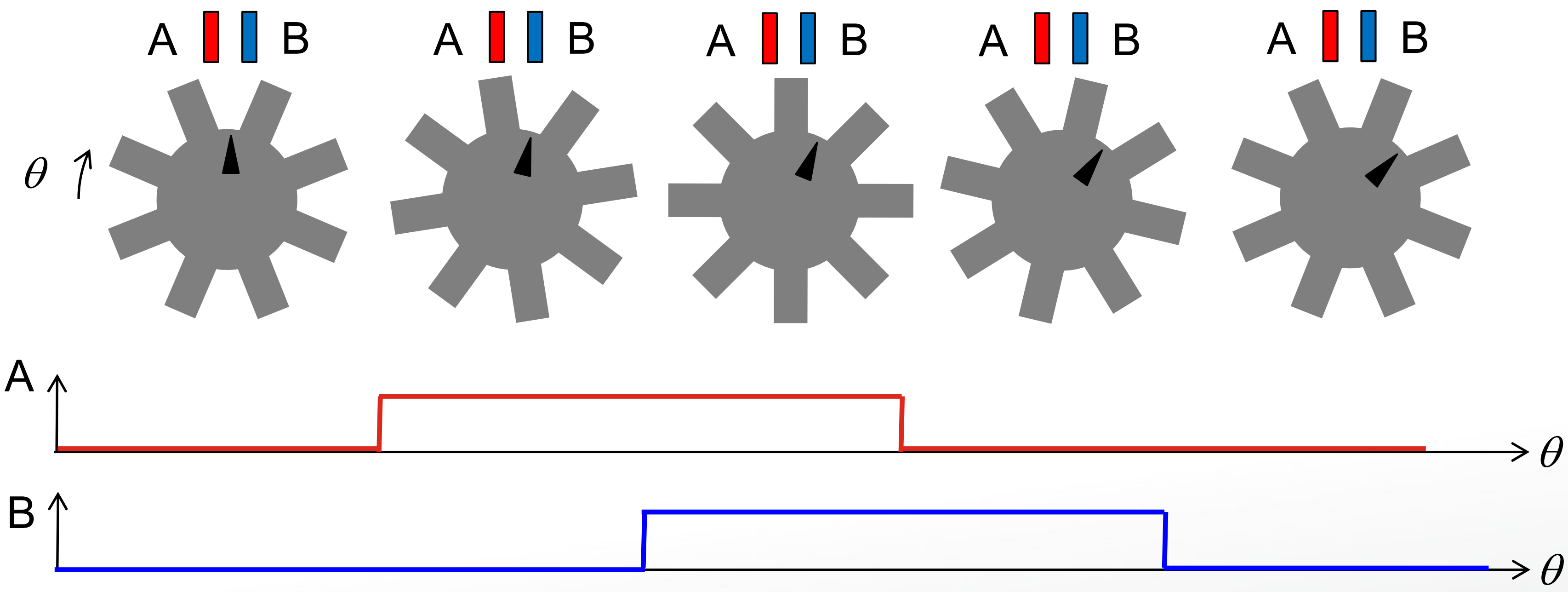


How do Encoders Work

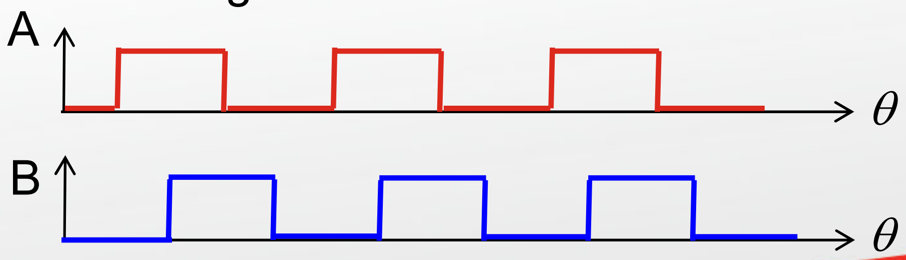
How do Encoders Work

- Let's first look at the optical encoder:

Clockwise



- On a larger scale:



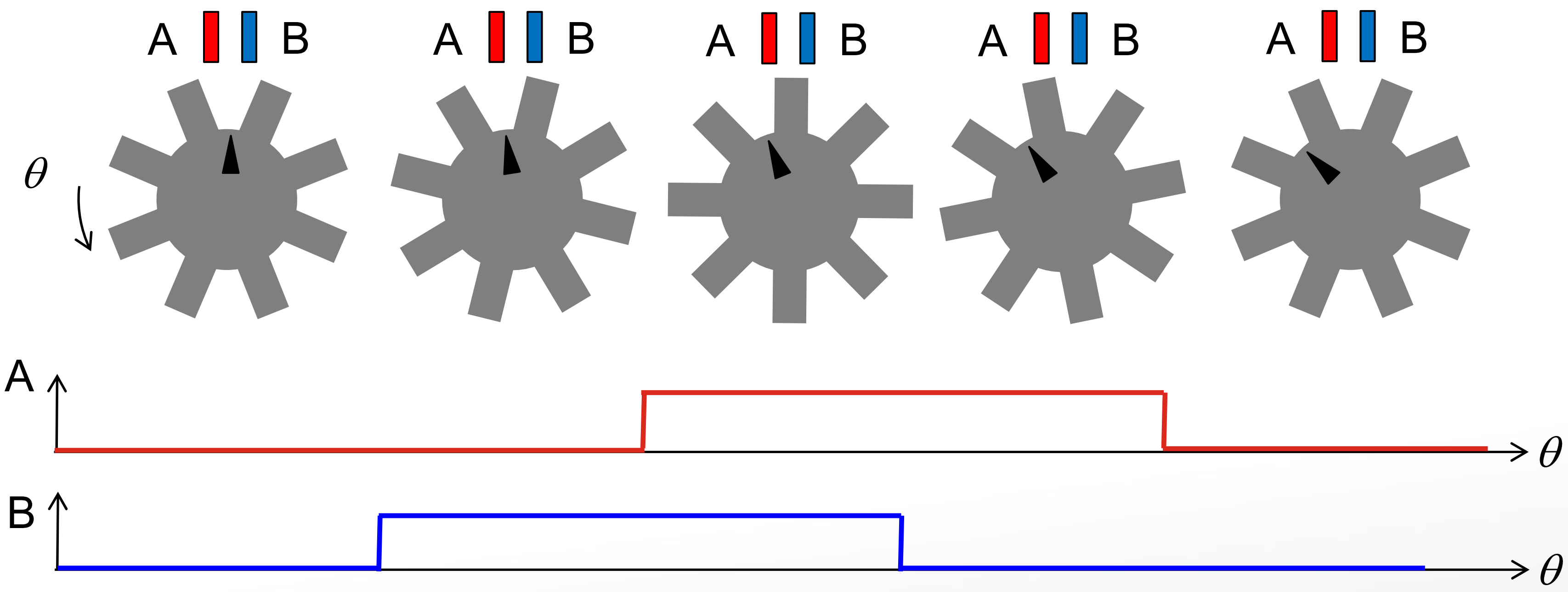
For CW rotation, B is delayed by $\underbrace{\text{¼ cycle}}$

"Quadrature"

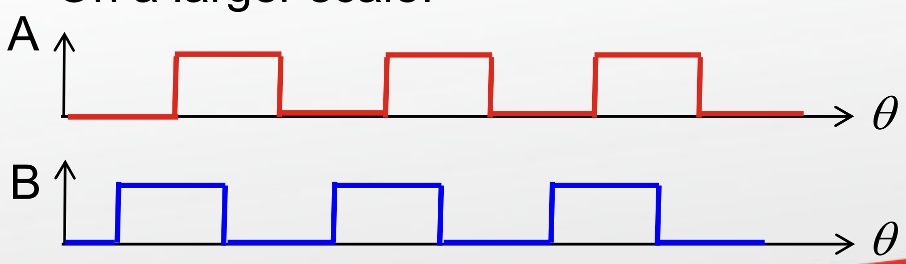
How do Encoders Work

- Let's first look at the optical encoder:

Counter-Clockwise



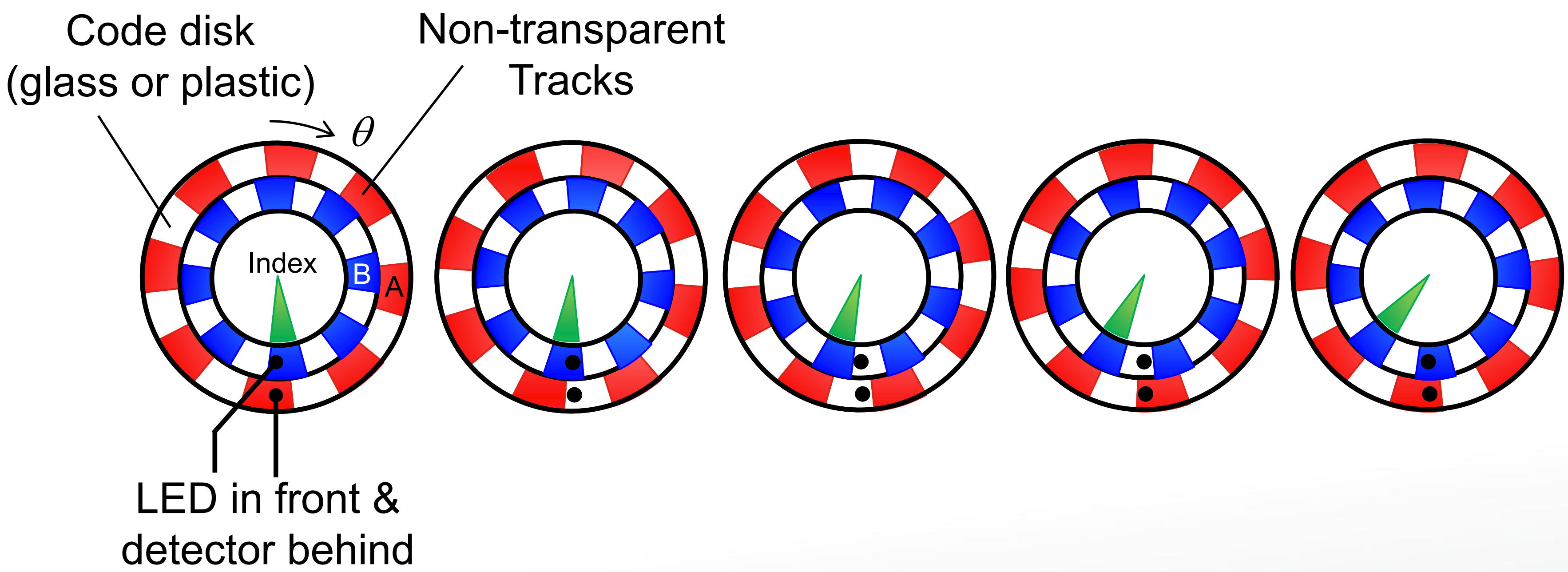
- On a larger scale:

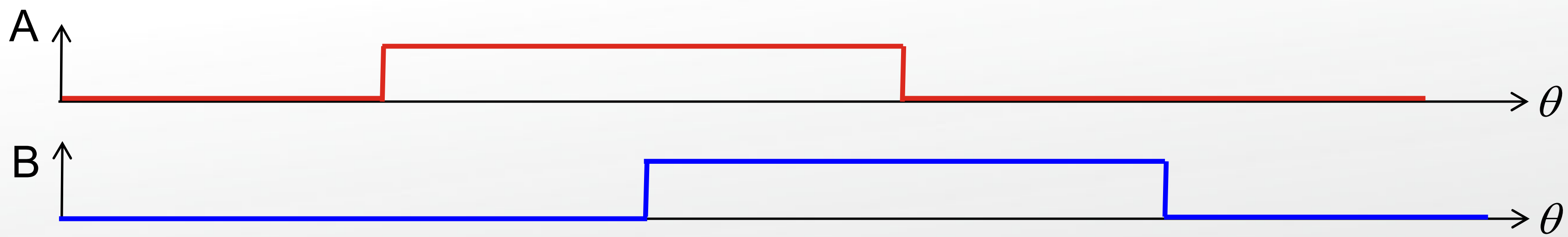


For CCW rotation, A is delayed $\underbrace{\text{by ¼ cycle}}_{\text{“Quadrature”}}$

How do Encoders Work

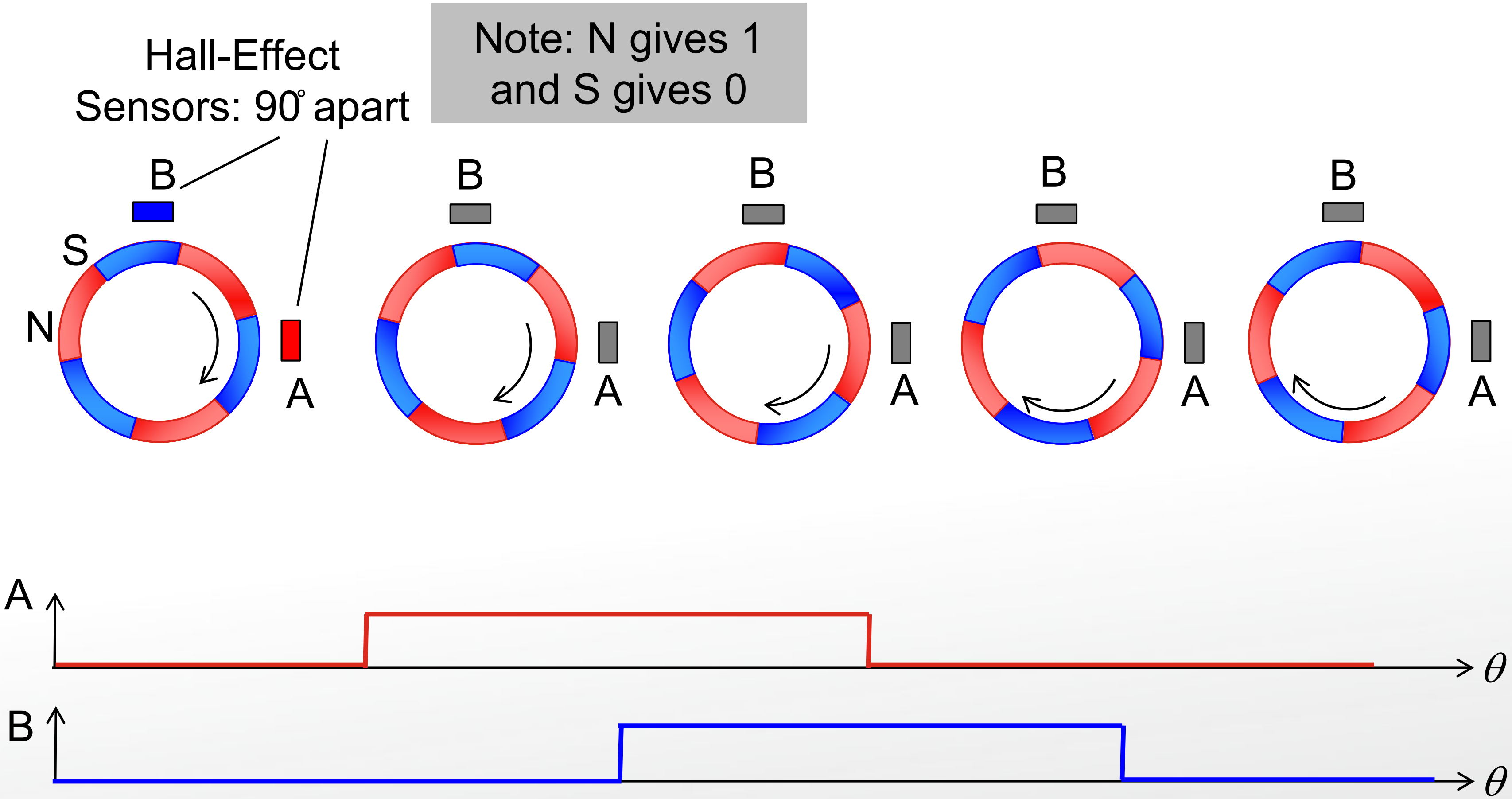
- Next let's look at another optical encoder: Clockwise





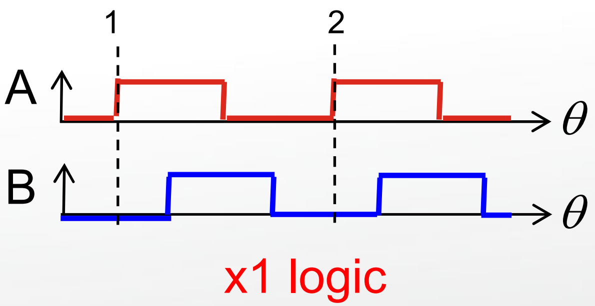
How do Encoders Work

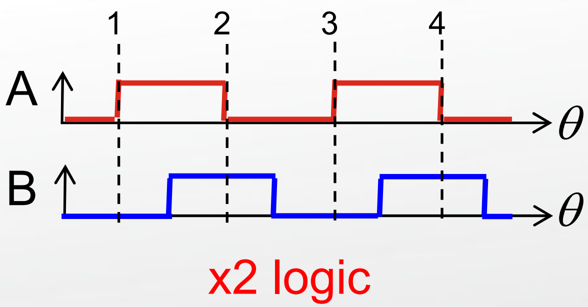
- Finally let's look at hall-effect encoder: Clockwise

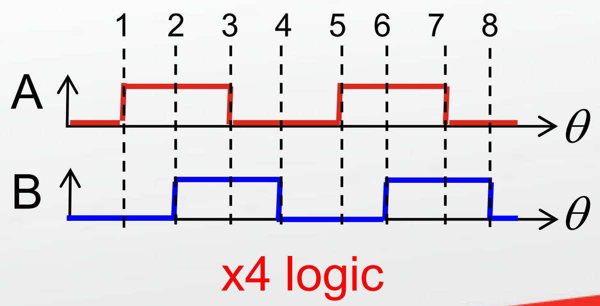


How to Use Pulse Signals

- How to determine distance travelled? – By counting the edges!
  - E.g. from datasheet, the encoder gives 100 PPR (Pulses per revolution).
    - If use only rising edge of channel A: 100 PPR → 3.6° per pulse.
    - If make use of both rising and falling edges on channel A, the resolution is increased by two (x2 logic): 3.6° / 2 = 1.8° per pulse.
    - If makes use of both rising and falling edges on both channels, the resolution is increased by four (x4 logic): 3.6° / 4 = 0.9° per pulse.
    - Cumulative sum of the counts = total distance travelled (Position).

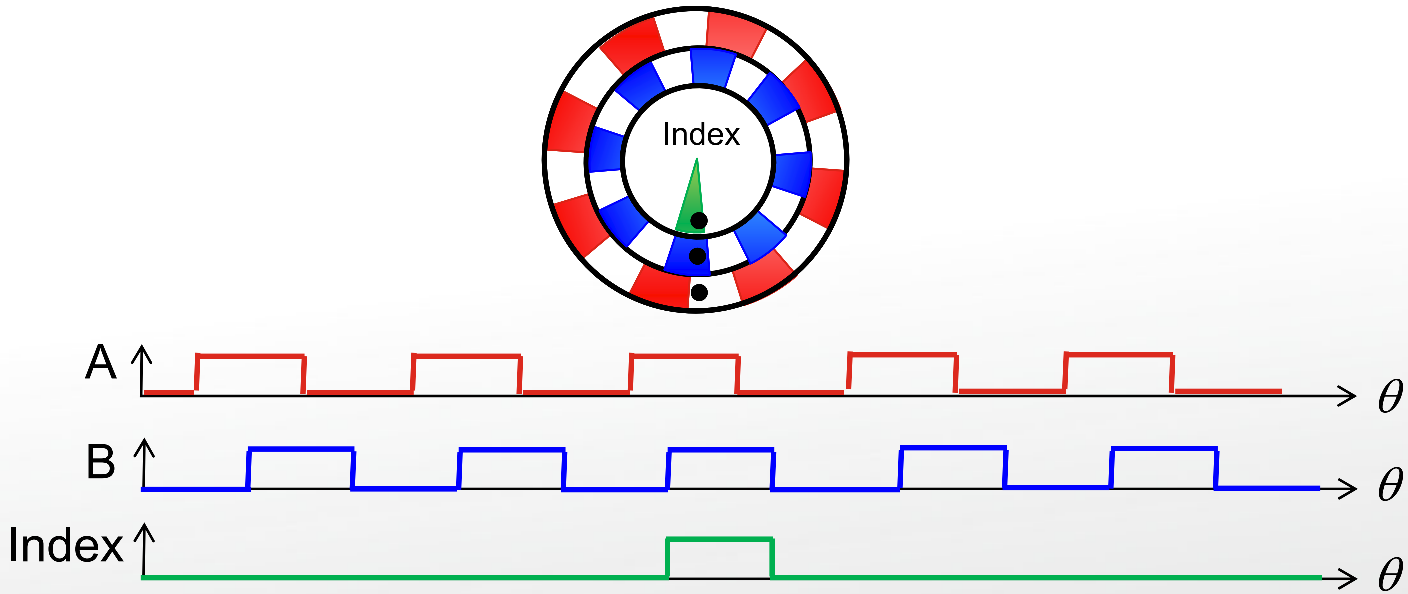






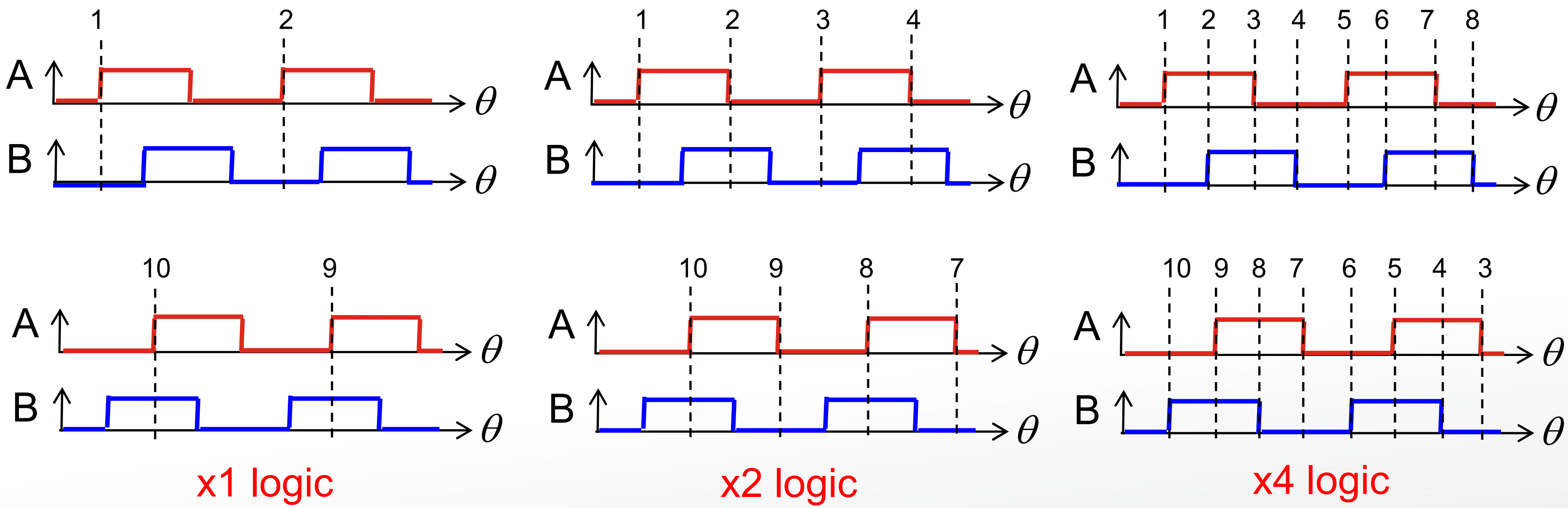
How to Use Pulse Signals

- How to determine absolute position?
  - Note: total distance travelled gives only the <u>relative</u> position from where we started counting, e.g. after shut down and restart of device.
  - To obtain an <u>absolute</u> position, often there is a third signal (called "Index") which yields only one pulse per revolution, which can be used as the zero position.



How to Use Pulse Signals

- How to determine direction? We know that:
  - for CW rotation, B is delayed.
  - For CCW rotation, A is delayed.

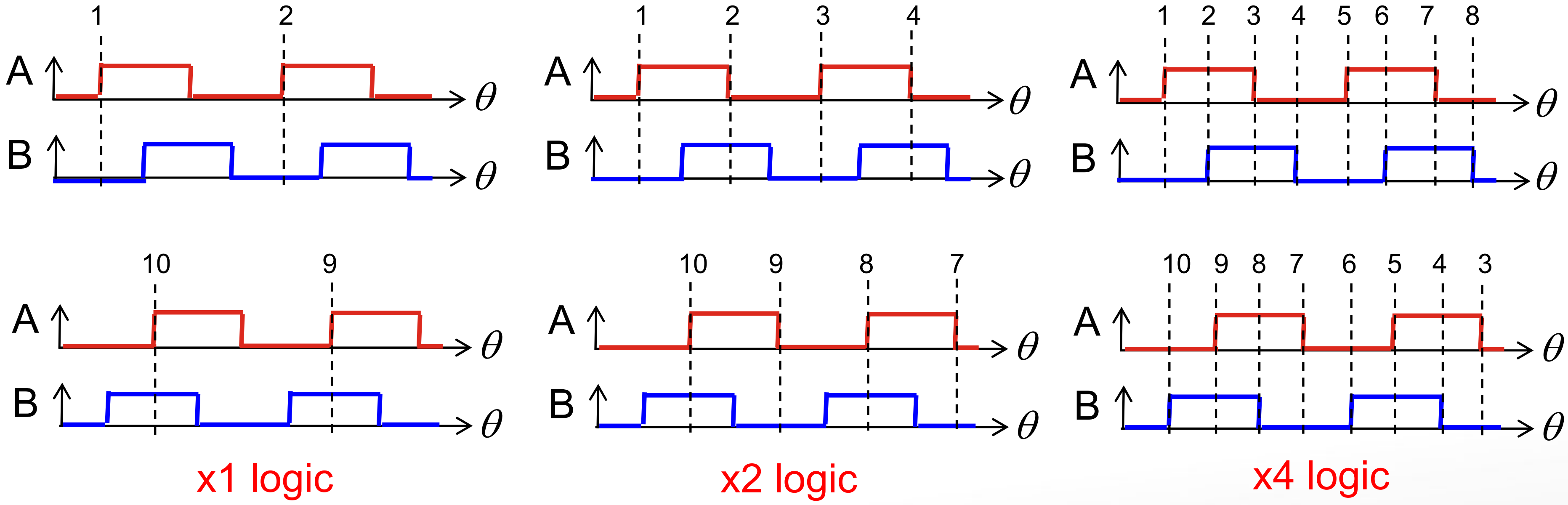


- Detect rising edge of one channel, and check status of the other channel.
  - E.g. in x1 logic:
  - A rises & B = 0 → CW
  - A rises & B = 1 → CCW

RMIT UNIVERSITY

How to Use Pulse Signals

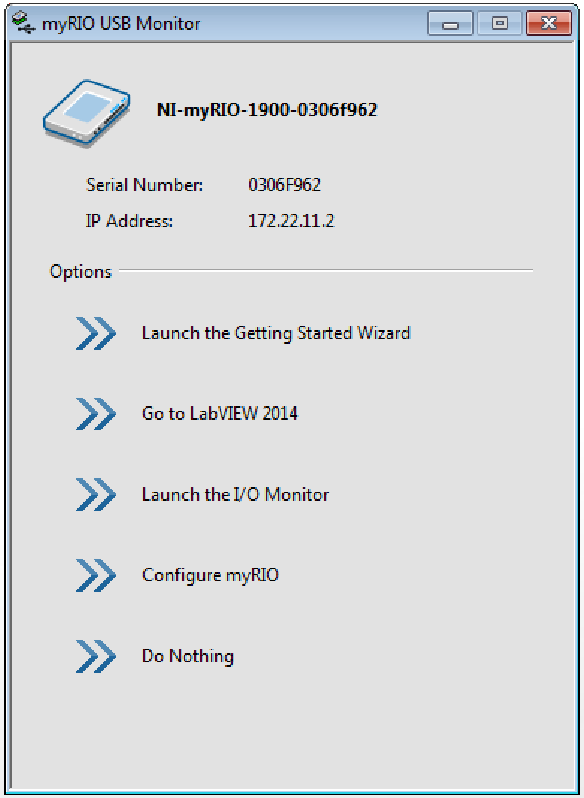
- What about x4 logic?



- At any of the rising and falling edge of A or B,
  - If A(now) != B(just now), then CW.
  - If A(now) = B(just now), then CCW.

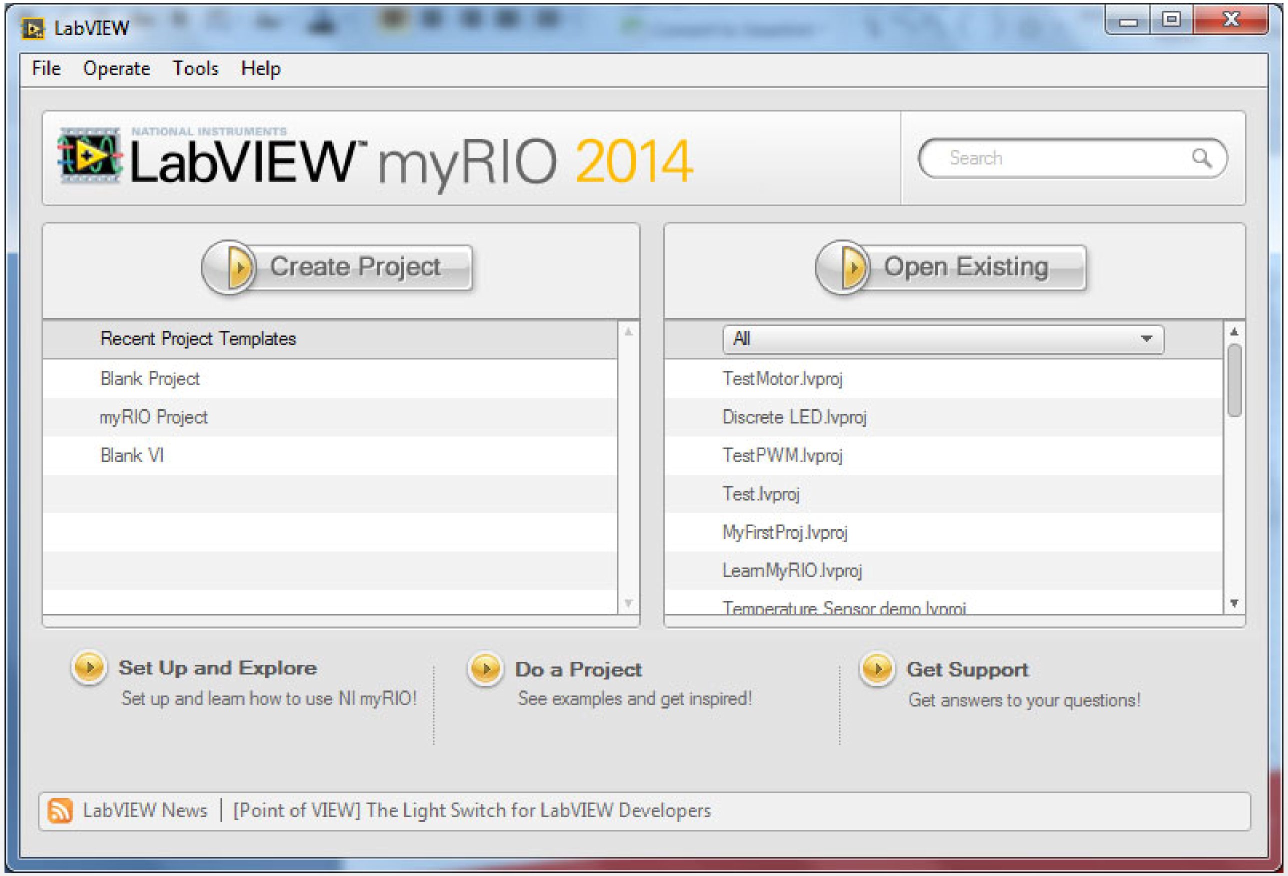
Use myRIO to read in Encoder

- In this section, we will create a VI to read and interpret the encoder signals.
- Connect myRIO to the PC via a USB cable.
- PC will automatically detect myRIO and the following dialogue will pop up:
  - Choose “Go to LabVIEW 201x”.



Use myRIO to read in Encoder

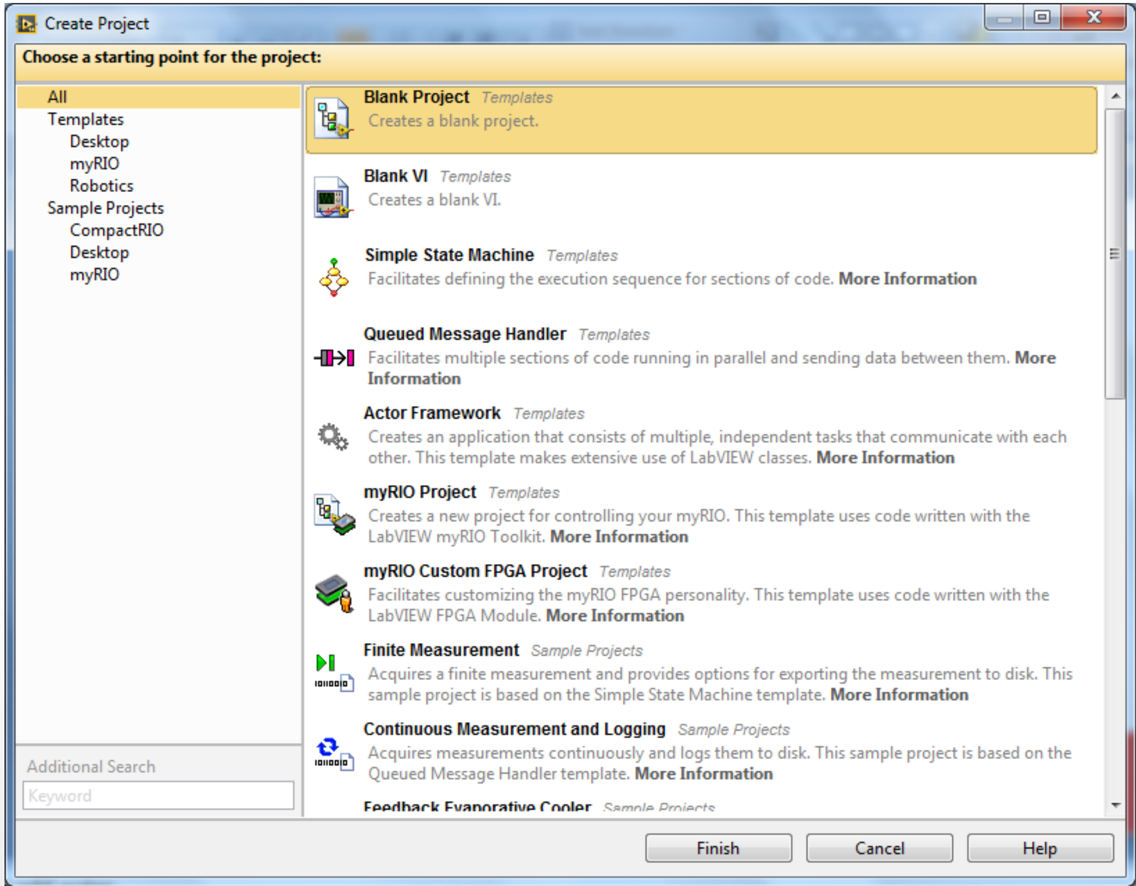
- LabVIEW will be started.



- Click on “Create Project”

Using Templates

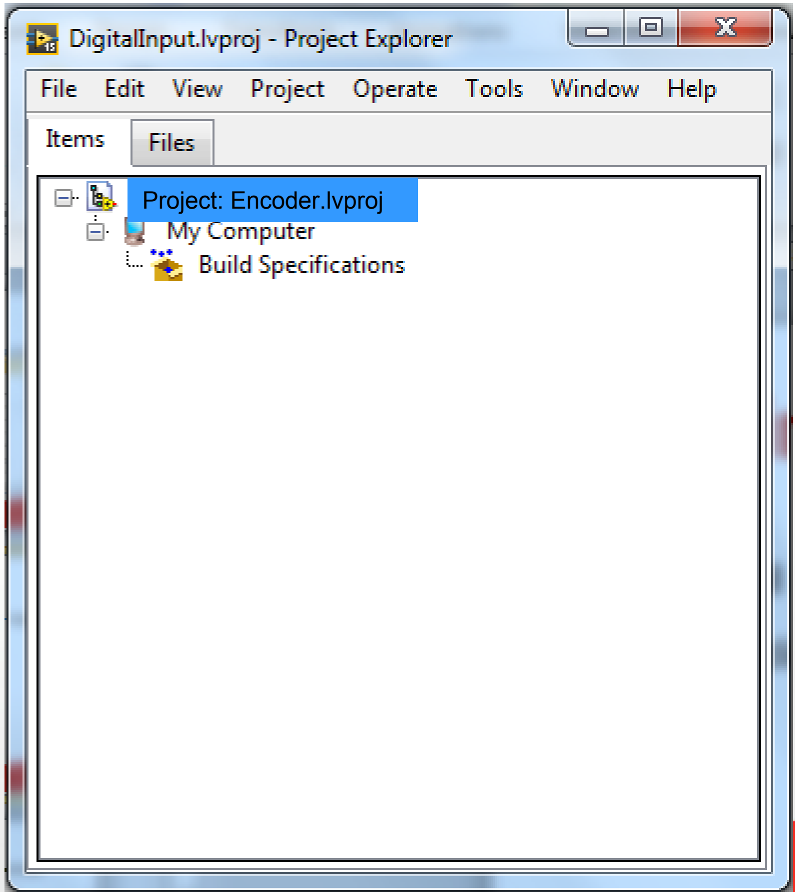
- There are templates for blank project, myRIO project and myRIO custom FPGA projects.



- Choose blank project.

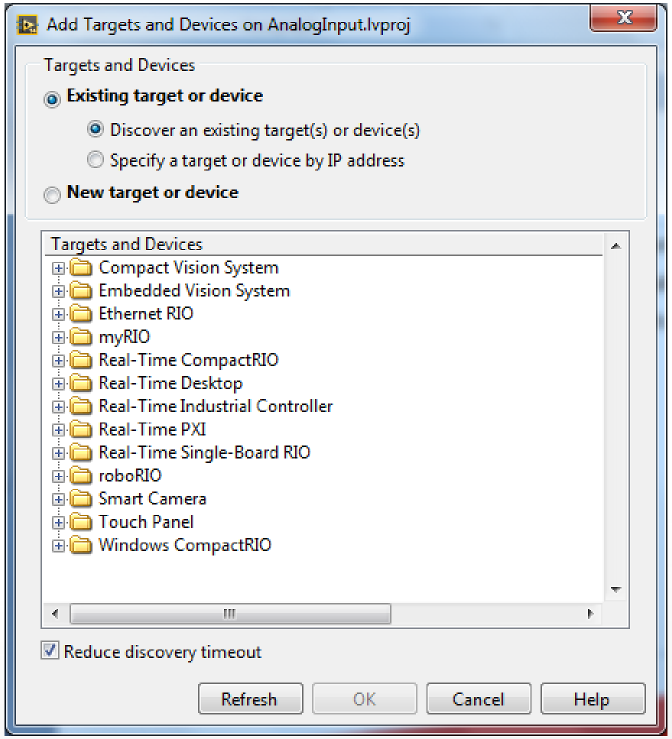
Setting up the Project

- You will see the project window.
- Right click on “Project: Untitled Project 1” → Save as → “Encoder”.



Setting up the Project

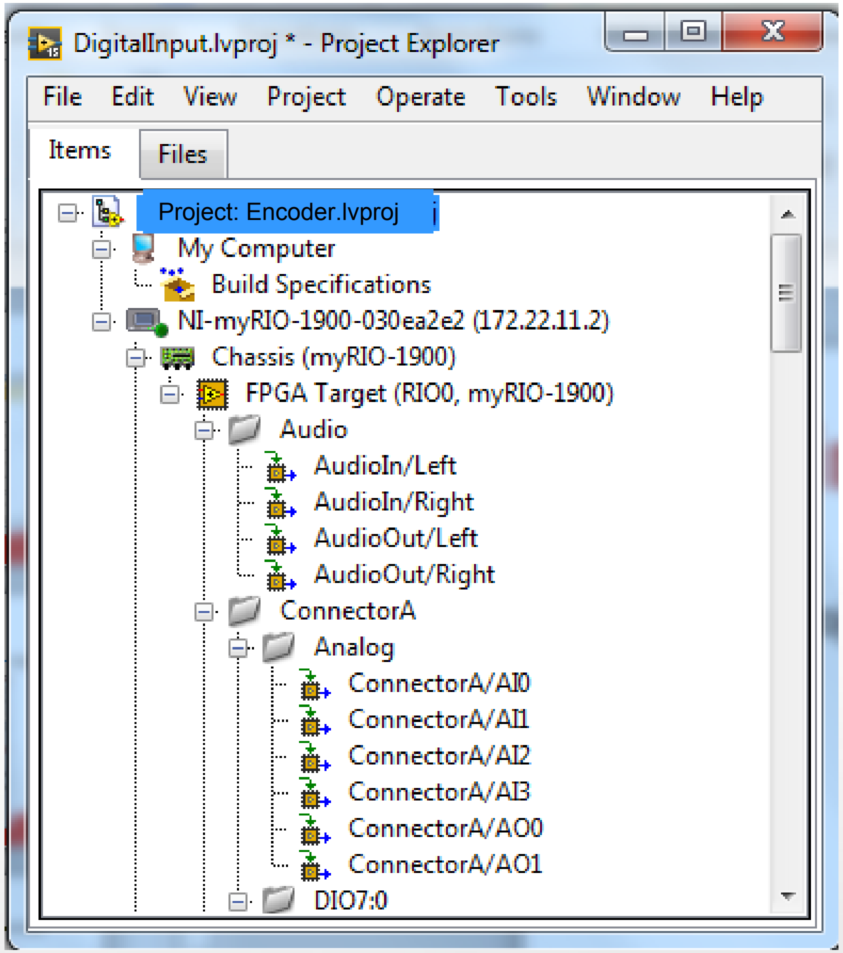
- Right click on Project Encoder → New → Targets and Devices



- Existing target or device:
  - Discover an existing target or device.
  - Or specify a target or device by IP address. (myRIO's IP is 172.22.11.2)
- Click on the "+" sign before myRIO.

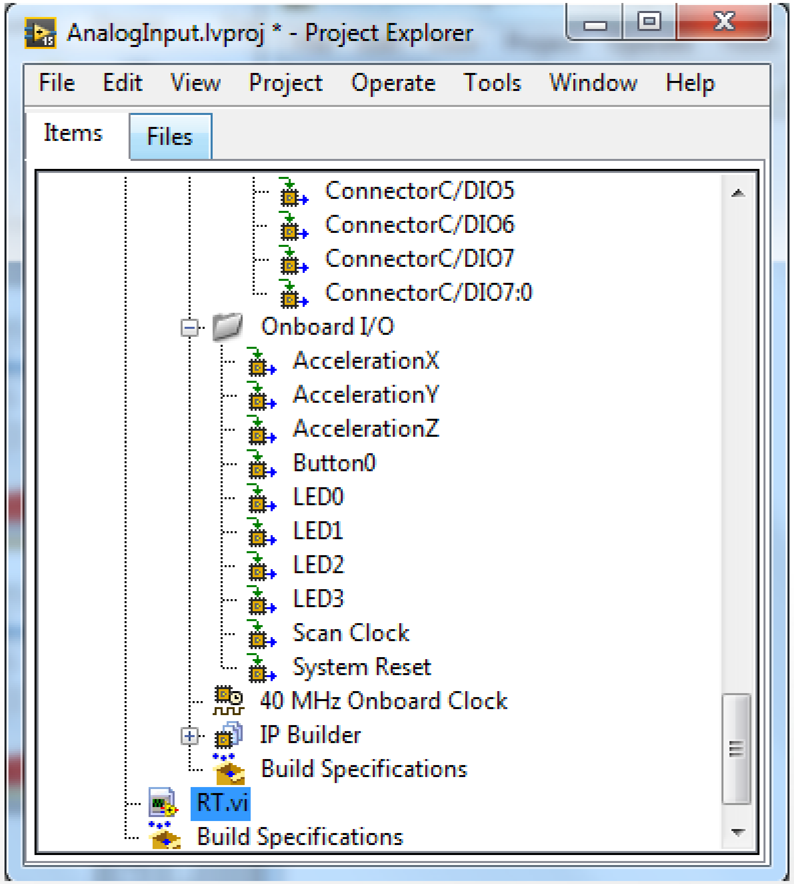
Setting up the Project

- The project tree now looks like this:



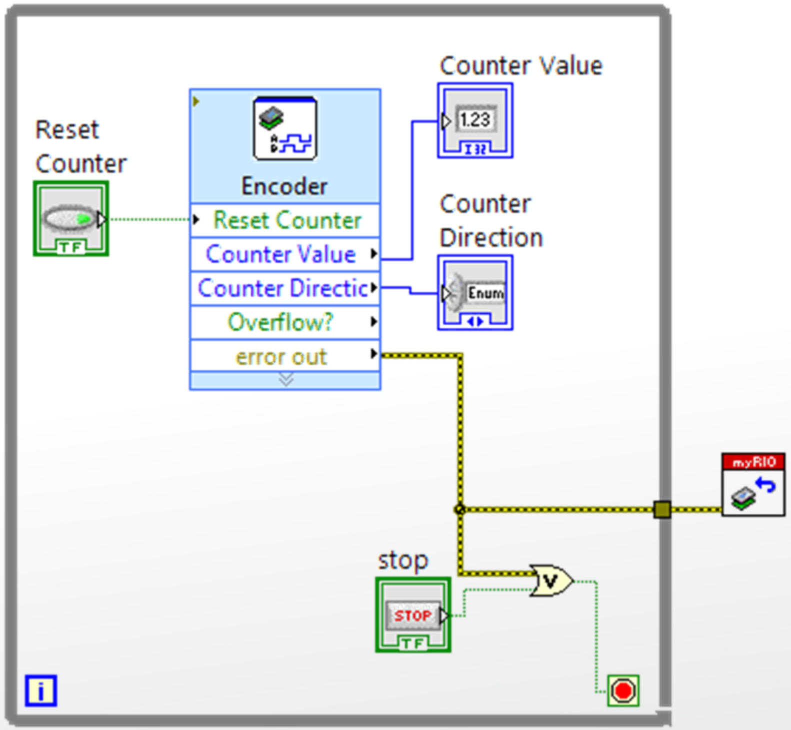
Setting up the Project

- Right click on “NI-myRIO-1900…” → New → VI
- A VI will open.
- Save it as RT.vi.
- Now we are ready to program the VI for reading digital inputs.



Reading Encoder Signals

- Recall that encoders give digital pulses which can be used for calculation of relative position, and also for determining the direction of motion.
- LabVIEW provides an FPGA function for interpreting the encoder signals.
- Let's create this VI:



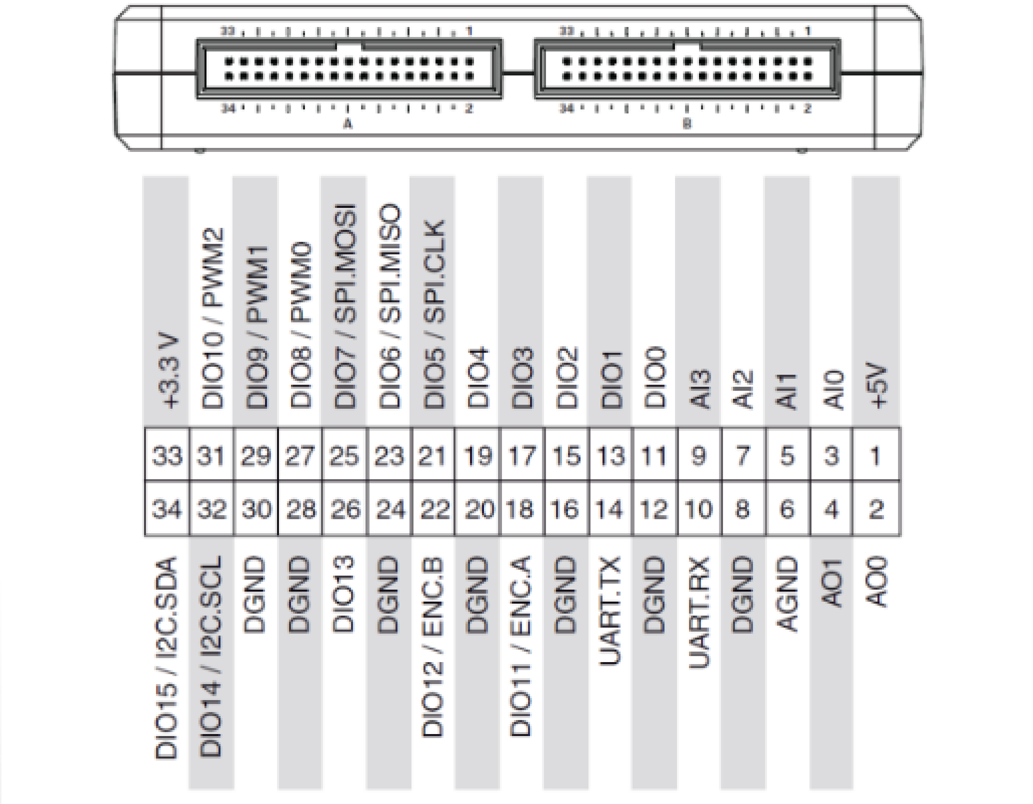
Reading Encoder Signals

- Your motor (RobotGear SKU-003141) has an integrated encoder.
- It also has 6 wires which are used as follows:

Colour	Meaning
Red	Motor +
Black	Motor -
Green	Encoder GND
Blue	Encoder Vcc (e.g. 5V)
Yellow	Encoder A Output
White	Encoder B Output

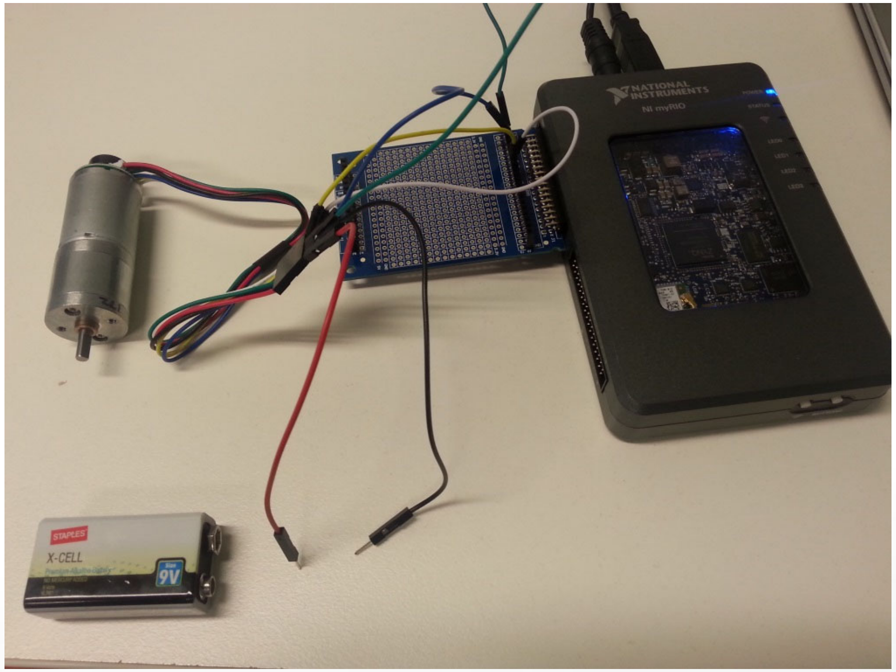
Reading Encoder Signals

- myRIO's datasheet shows the following port configuration:



Reading Encoder Signals

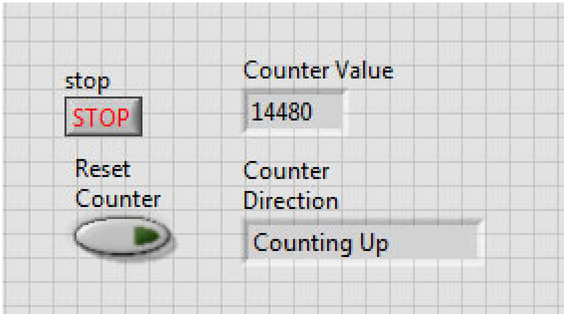
- Therefore, please connect your motor with myRIO as follows:



Motor wire	Connect to
Red	Battery +
Black	Battery -
Green	myRIO GND
Blue	myRIO 5V
Yellow	myRIO DIO 11
White	myRIO DIO 12

Reading Encoder Signals

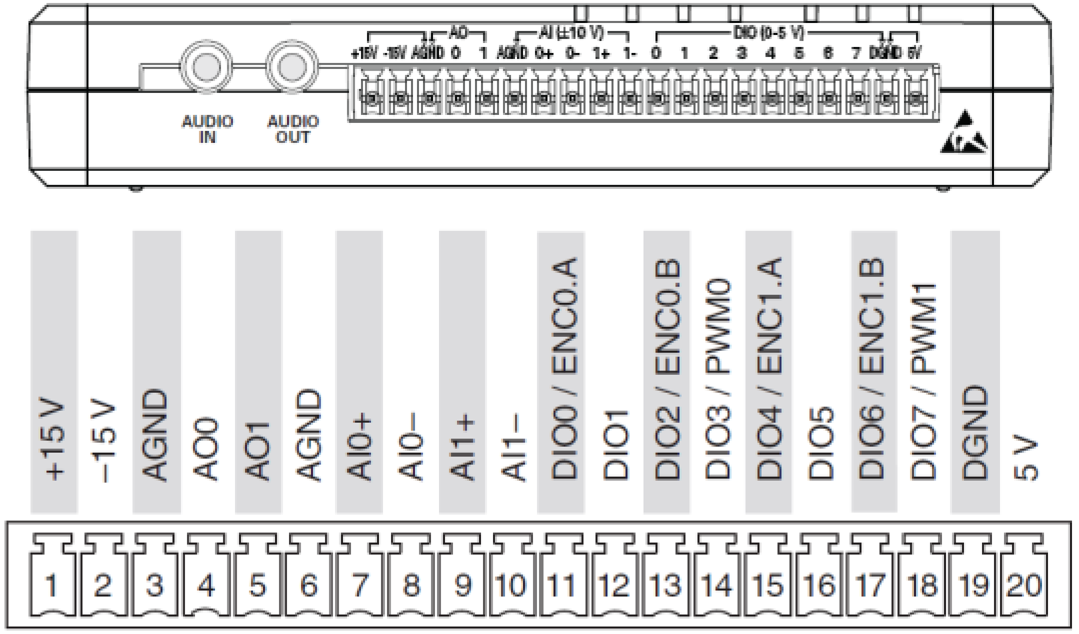
- Run the LabVIEW Program and you should see the following result:



- The code shows the counter value, as well as the direction.
- You can also press reset to bring the counter back to zero.

Writing Your Own Encoder Codes

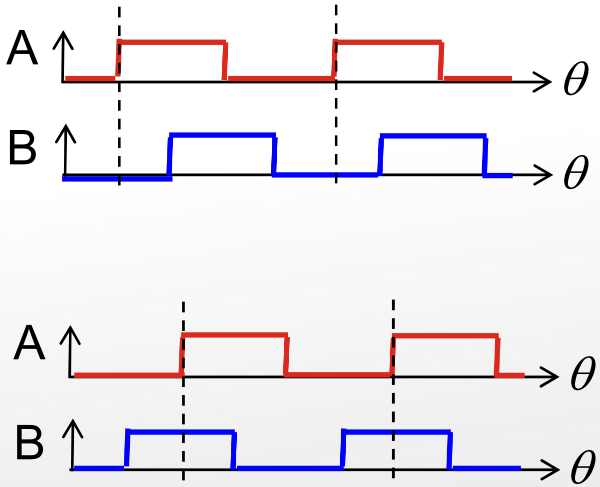
- Using myRIO's built-in encoder function, a total of four encoders can be supported:
  - One at MXP A
  - One at MXP B
  - Two at MSP C



- If you need more than 4 encoders in any other mechatronics / robotics project, then you need to write your own encoder code.
- Let's do this.

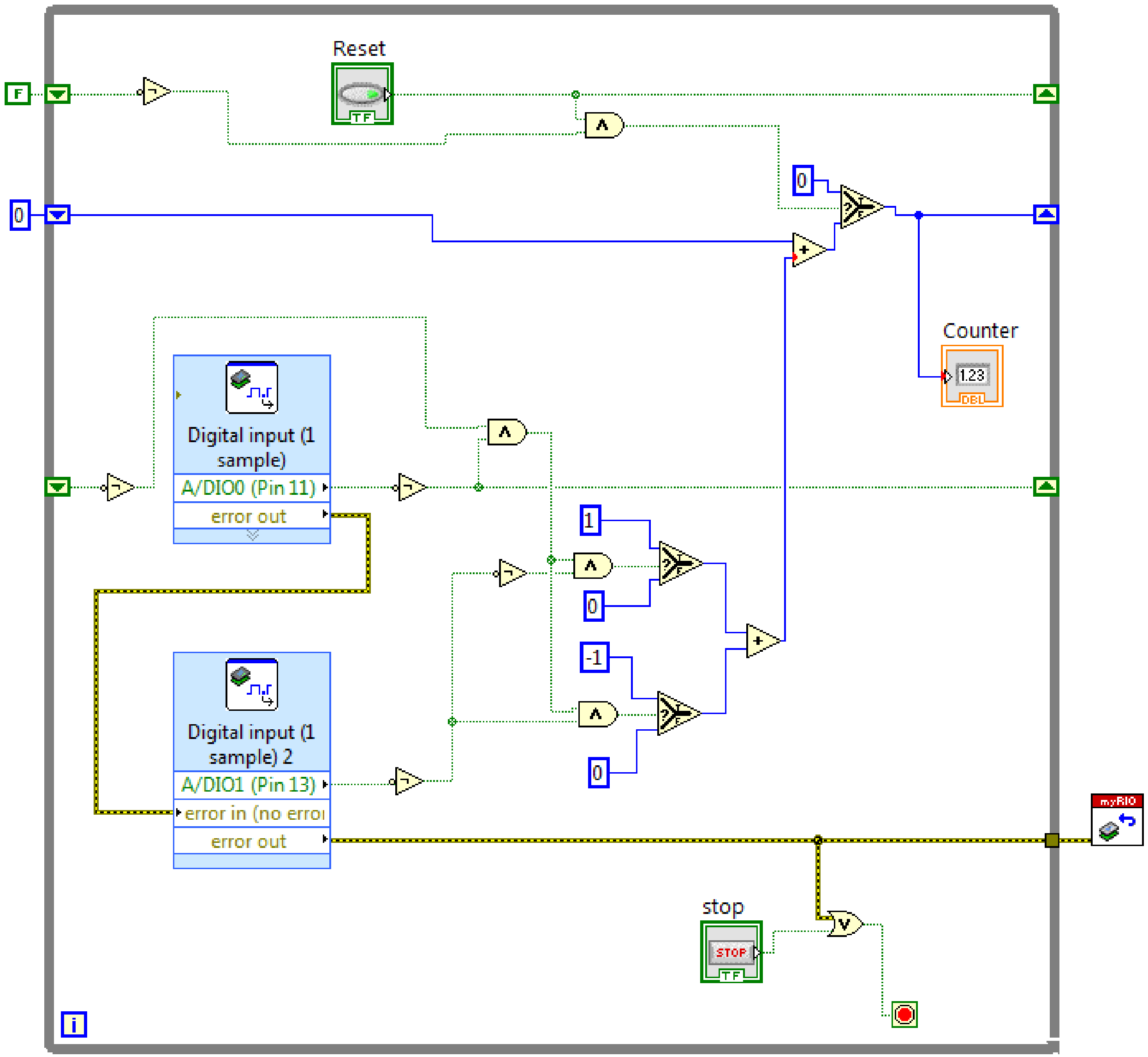
Writing Your Own Encoder Codes

- Firstly, remember that encoder signals are just digital signals.
- So we can read signals A and B through any of the digital inputs.
- The next question is how to process the signals to do the counting.
- Start with x1 logic first:
  - Detect rising edge in A, and check status of B.
    - A rises & B = 0 → CW
    - A rises & B = 1 → CCW

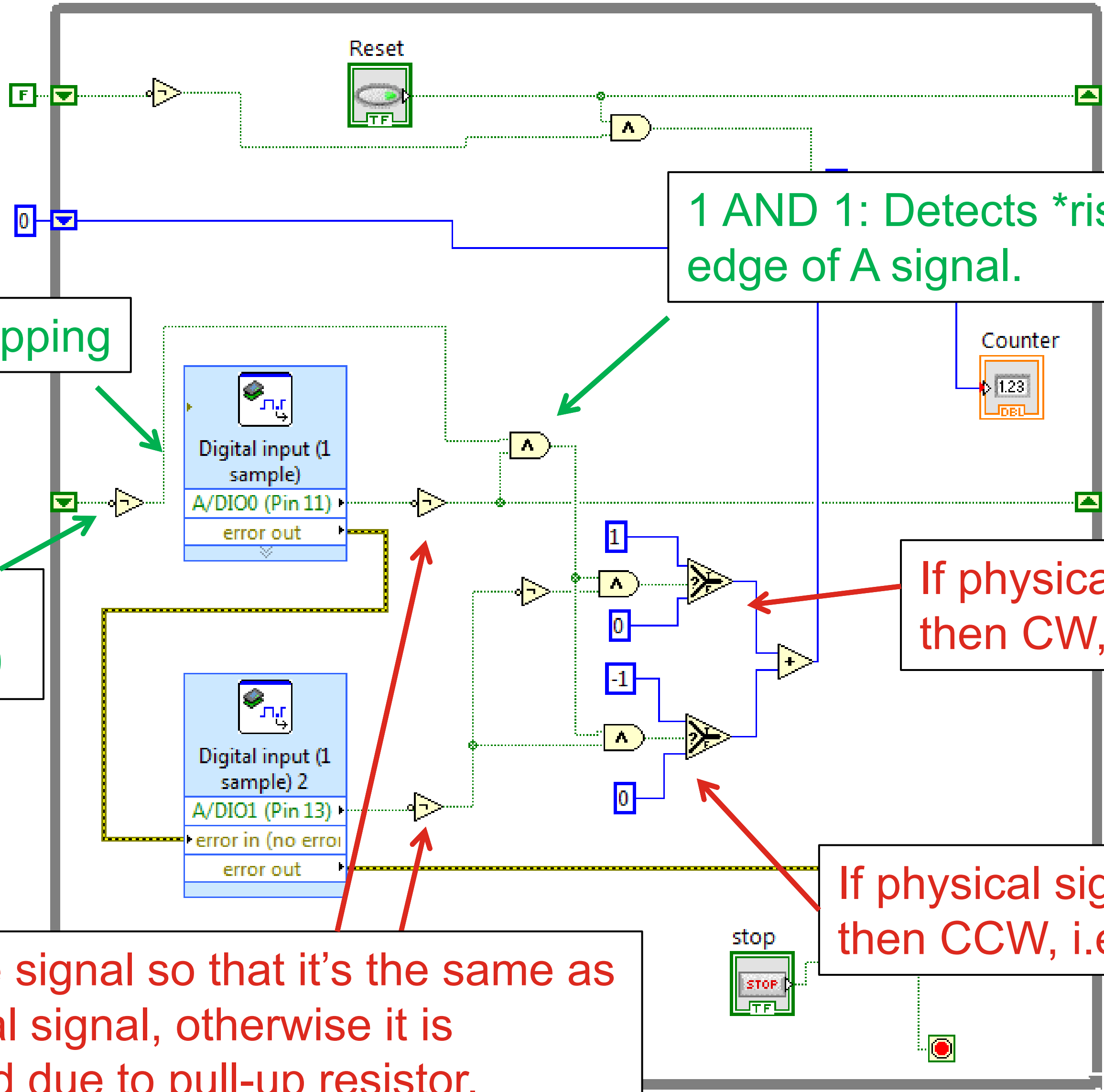


x1 logic

Writing Your Own Encoder Codes



Writing Your Own Encoder Codes



1 AND 1: Detects *rising* edge of A signal.

1 after flipping

Assume just now = 0, and now = 1

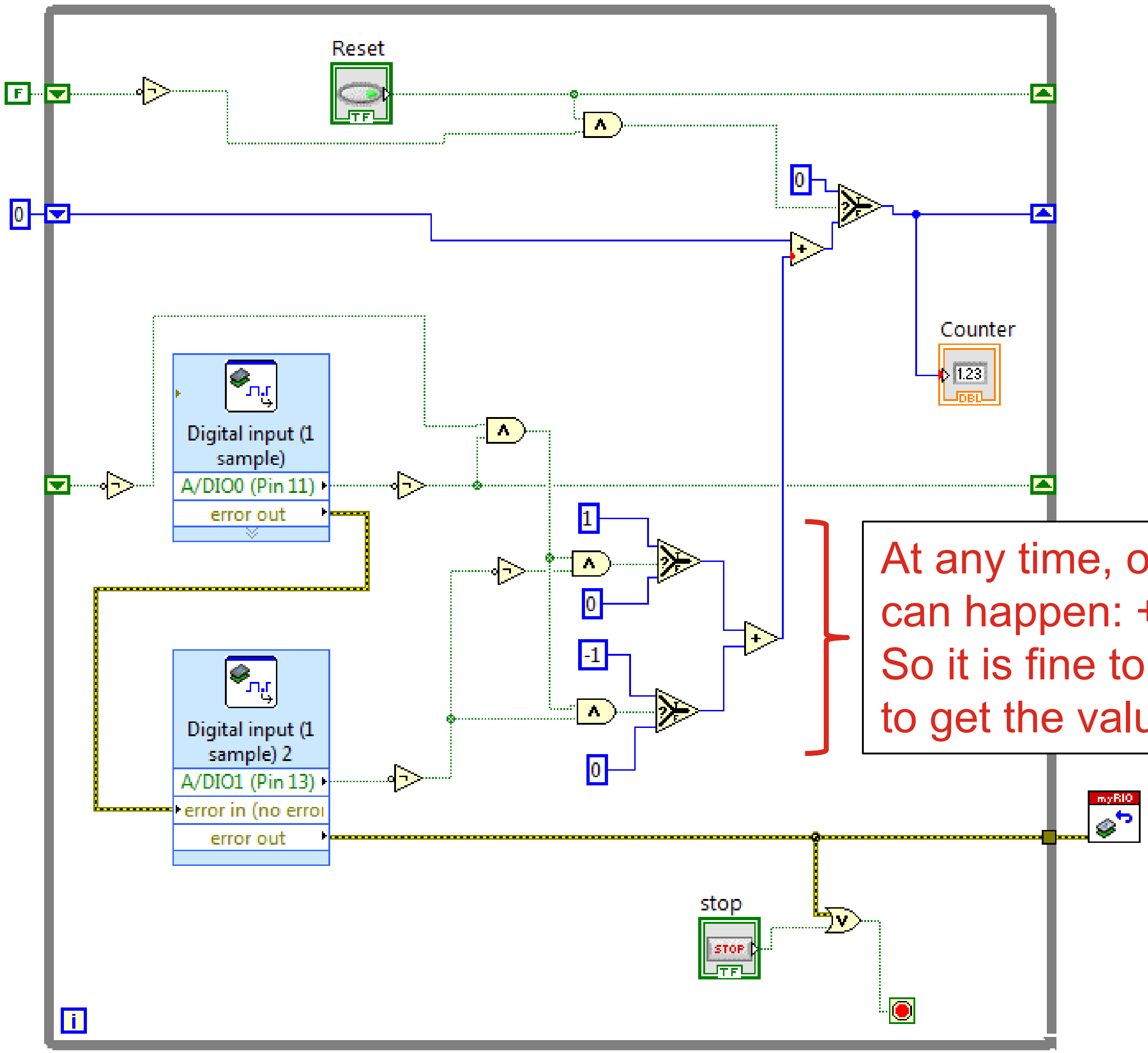
Just now 0

If physical signal of B is 0, then CW, i.e. count +1.

If physical signal of B is 1, then CCW, i.e. count -1.

Flip the signal so that it's the same as physical signal, otherwise it is inverted due to pull-up resistor.

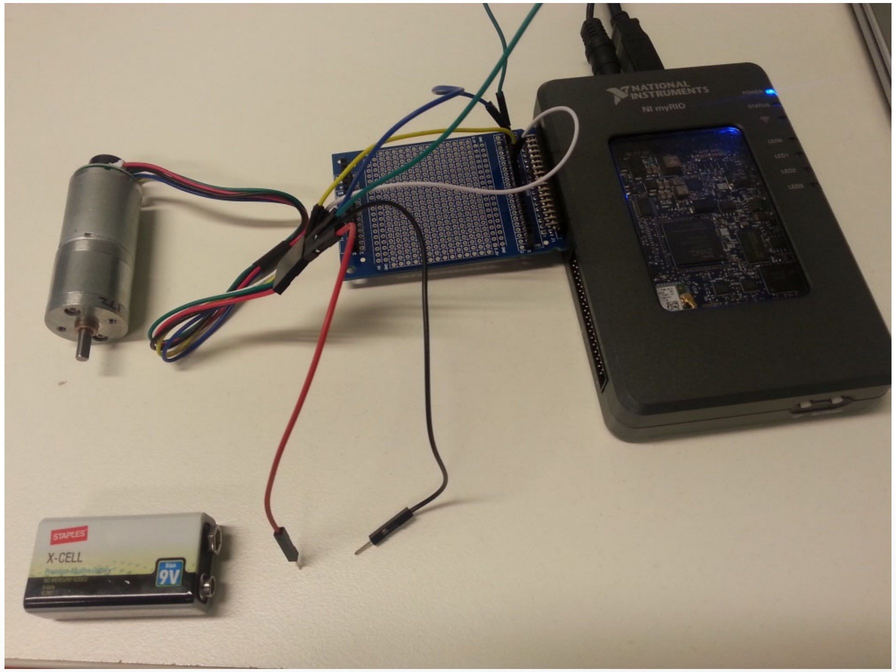
Writing Your Own Encoder Codes



At any time, only one of these can happen: +1, -1 or 0.
So it is fine to use summation to get the value.

Writing Your Own Encoder Codes

- Connect your motor with myRIO as follows:

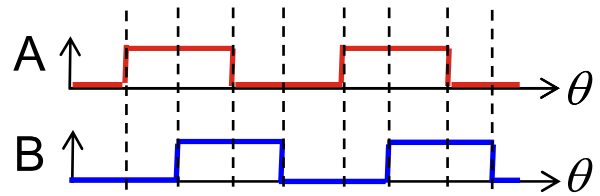


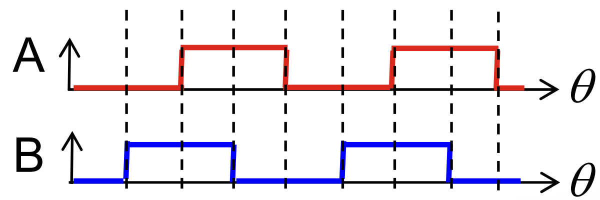
Motor wire	Connect to
Red	Battery +
Black	Battery -
Green	myRIO GND
Blue	myRIO 5V
Yellow	myRIO DIO 0
White	myRIO DIO 1

Depending on your code

Writing Your Own Encoder Codes

- If you are free, you may also try writing your own codes for the x4 logic:





x4 logic

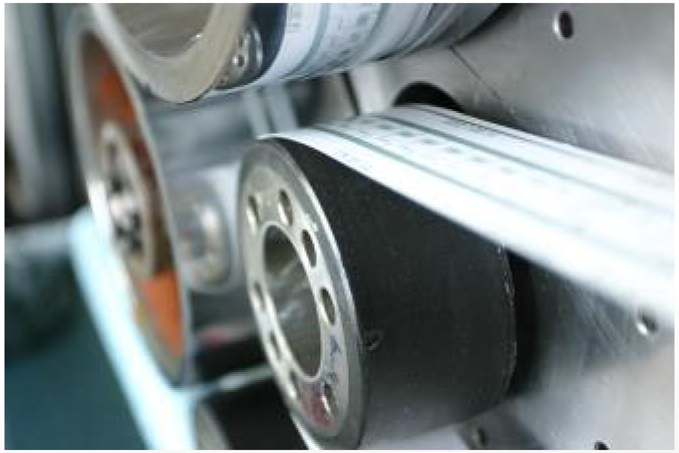
- At any of the rising and falling edge of A or B,
  - If A(now) != B(just now), then CW.
  - If A(now) = B(just now), then CCW.

Content

- Position Measurement
  - Measure using myRIO
- Speed Measurement
- Vibration and Acceleration Measurement
- Stress and Strain Measurement
  - Measure using myRIO
- Saving (Sensor) Data using myRIO
- Attachment: Writing FPGA Codes

Speed Sensors

- To measure the speed of an object.
- Usage examples:
  - Measure the speed of DC motor, AC motor etc.
    - For speed control of machines etc.
  - Measure the speed of automated forklift.

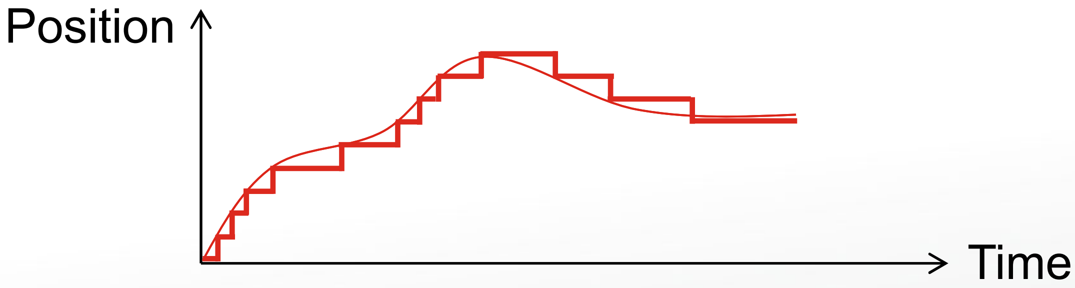


Machine for printed electronics
https://commons.wikimedia.org/wiki/File:Elektronikdruck.jpg

Automated Forklift
https://commons.wikimedia.org/wiki/File:Forklift_AGV_with_Straddle,_courtesy_of_Egemin_Automation_Inc..jpg

Speed Sensors (1)

- Encoder + derivation
  - This is one of the most widely used methods in Mechatronics.
  - Speed is defined as $\dfrac{\text{Change in Position}}{\text{Change in Time}} = \dfrac{\Delta p}{\Delta t}$
  - Let's first look at the an example of position signal, which is the cumulative sum of the edge counts:



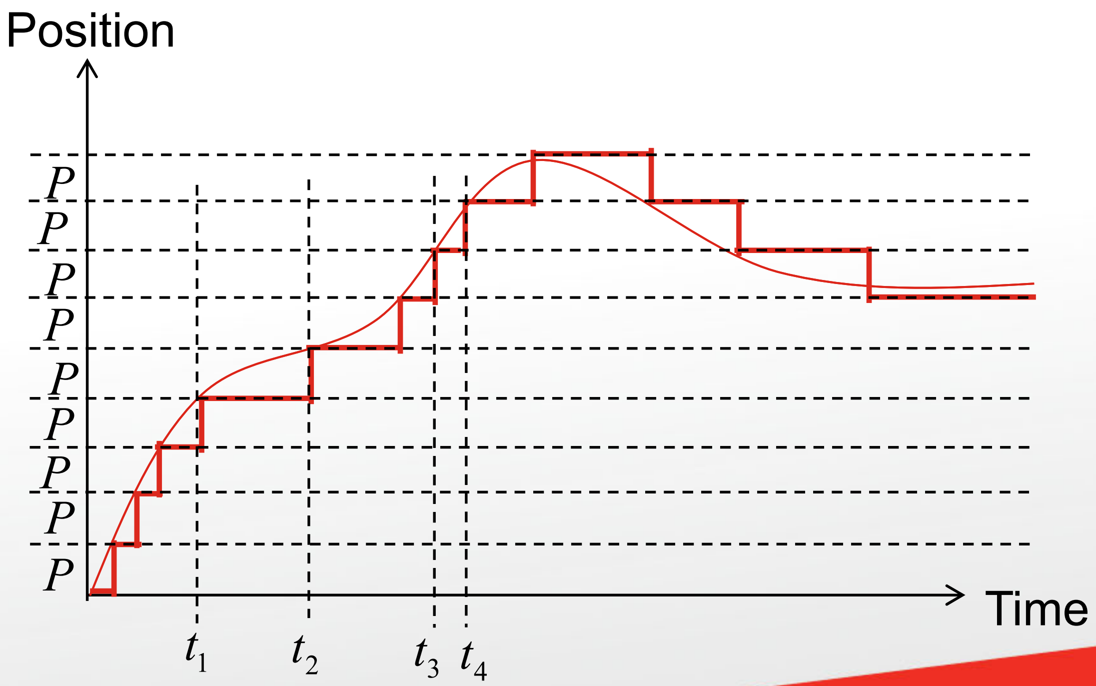
- The position signal is discontinuous due to quantization → resolution of the encoder. Example: every level increase is 0.9°

Deriving Speed from Position

- There are two ways to calculate speed from position:

a) Make use of the constant quantization level P:

- Determine the time duration between position increment.
- Then $\text{Speed}_1 = \dfrac{\Delta p}{\Delta t} = \dfrac{P}{t_2 - t_1}$ $\text{Speed}_2 = \dfrac{\Delta p}{\Delta t} = \dfrac{P}{t_4 - t_3}$ → constant



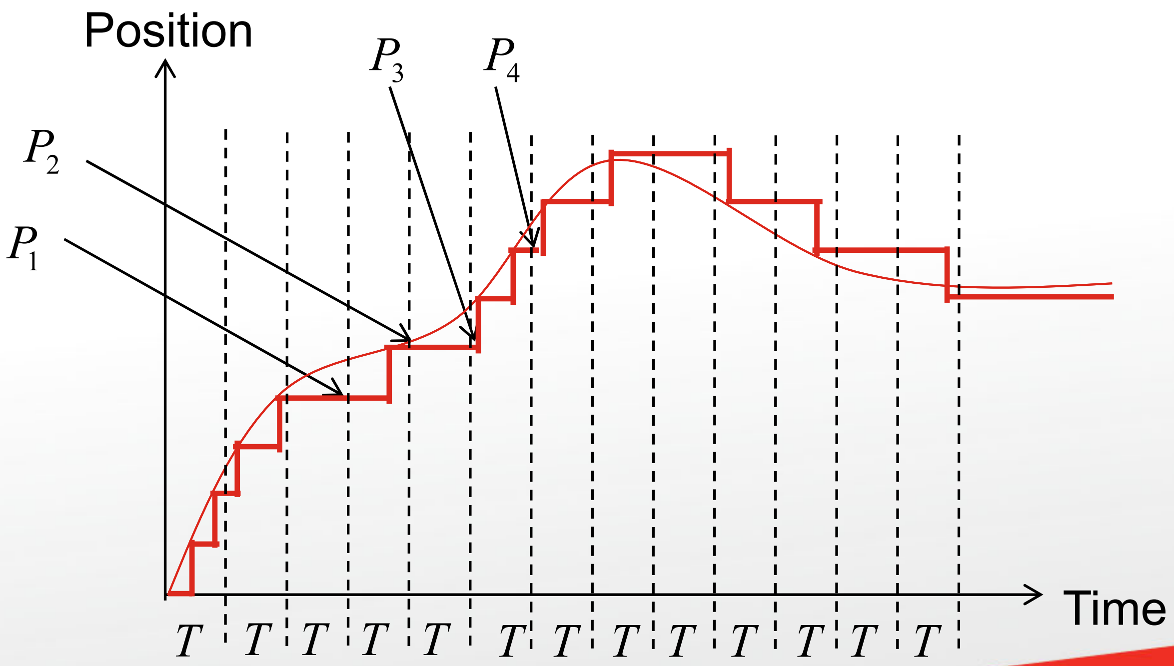
a)
Good for low speed

Deriving Speed from Position

- Second method:

b) Make use of the constant sampling interval T of the discrete-time system.

- At the sampling time instance, get the P value.
- Then $\text{Speed}_3 = \dfrac{\Delta p}{\Delta t} = \dfrac{P_2 - P_1}{T}$ $\quad$ $\text{Speed}_4 = \dfrac{\Delta p}{\Delta t} = \dfrac{P_4 - P_3}{T}$ → constant



b)
Good for
high speed

Content

- Position Measurement
  - Measure using myRIO
- Speed Measurement
- Vibration and Acceleration Measurement
- Stress and Strain Measurement
  - Measure using myRIO
- Saving (Sensor) Data using myRIO
- Attachment: Writing FPGA Codes

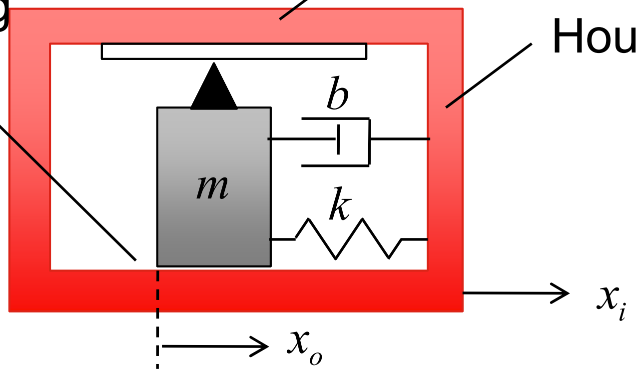
Vibration and Acceleration Measurement

- To measure the vibration and acceleration of the device.
- Usage examples:
  - To detect sudden impact, e.g. Accelerometer detects hard disk drives fall towards ground and commands the read-write head to retract, preventing catastrophic scratch on the media surface.
  - Accelerometer can also identifies the direction of gravitational force, therefore used in smart phones for orientation of screen.

Hard Disk Drive Head Crash
https://commons.wikimedia.org/wiki/File:Hard_disk_head_crash.jpg

Smart Phones
https://commons.wikimedia.org/wiki/File:Samsung_Galaxy_S6_edge%2B.jpg

Accelerometer



- When the housing (attached to external object) accelerates, there is a relative motion between housing and the seismic mass.

$$x_r = x_i - x_o$$

- This relative motion is measured by the displacement transducer.
- Spring force: $F_k = k(x_i - x_o) = kx_r$
- Damping force: $F_b = b(\dot{x}_i - \dot{x}_o) = b\dot{x}_r$
- Newtons law for seismic mass: $m\ddot{x}_o = \sum F_{external}$
- Therefore: $m\ddot{x}_o = F_k + F_b = kx_r + b\dot{x}_r$
- Which gives: $m(\ddot{x}_i - \ddot{x}_r) = kx_r + b\dot{x}_r$
- Or: $m\ddot{x}_i = m\ddot{x}_r + kx_r + b\dot{x}_r$
- Equivalently: $\ddot{x}_r + \dfrac{b}{m}\dot{x}_r + \dfrac{k}{m}x_r = \ddot{x}_i$

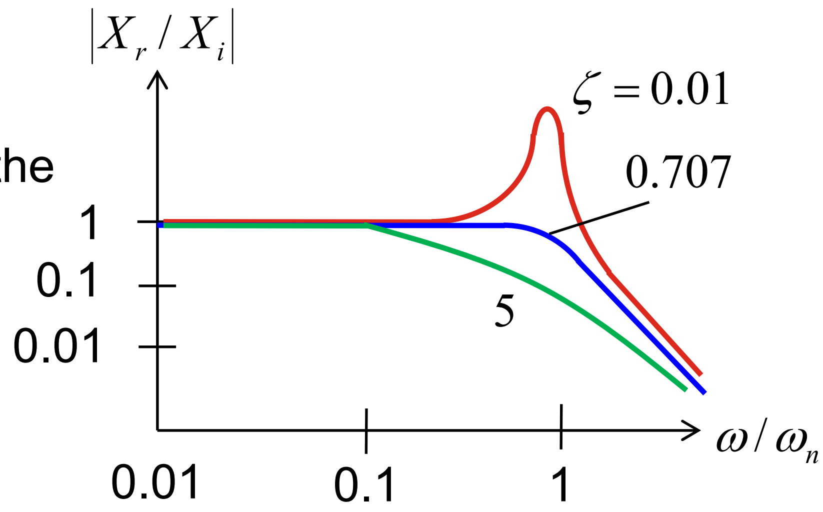
Accelerometer

- The previous equation can be written in the standard 2nd order system equation:

$$\ddot{x}_r + 2\zeta\omega_n\dot{x}_r + \omega_n^2 x_r = \ddot{x}_i$$

- With natural frequency: $\omega_n = \sqrt{\frac{k}{m}}$

- And damping ratio: $\zeta = \frac{b}{2\sqrt{km}}$



Amplitude Response of 2nd Order System

- If the accelerometer is designed such that the damping ratio is 0.707, the gain will be one for large frequency range.

- Then, if we operate the accelerometer within its bandwidth, we will have:

$$\omega_n^2 x_r = \ddot{x}_i$$

- That means, by measuring the relative displacement x_r, the acceleration of the housing (and thus object) can be calculated.

Displacement Transducer in Accelerometer

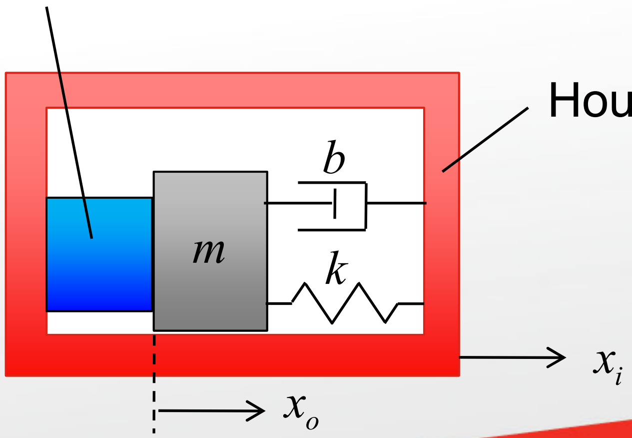
- There are various ways to measurement the relative motion in the accelerometer.

  a) Potentiometer:

    - Seismic mass attached to wiper arm of potentiometer.
    - Suitable for low frequency (<30Hz) vibration measurement.

  b) Piezoelectric:

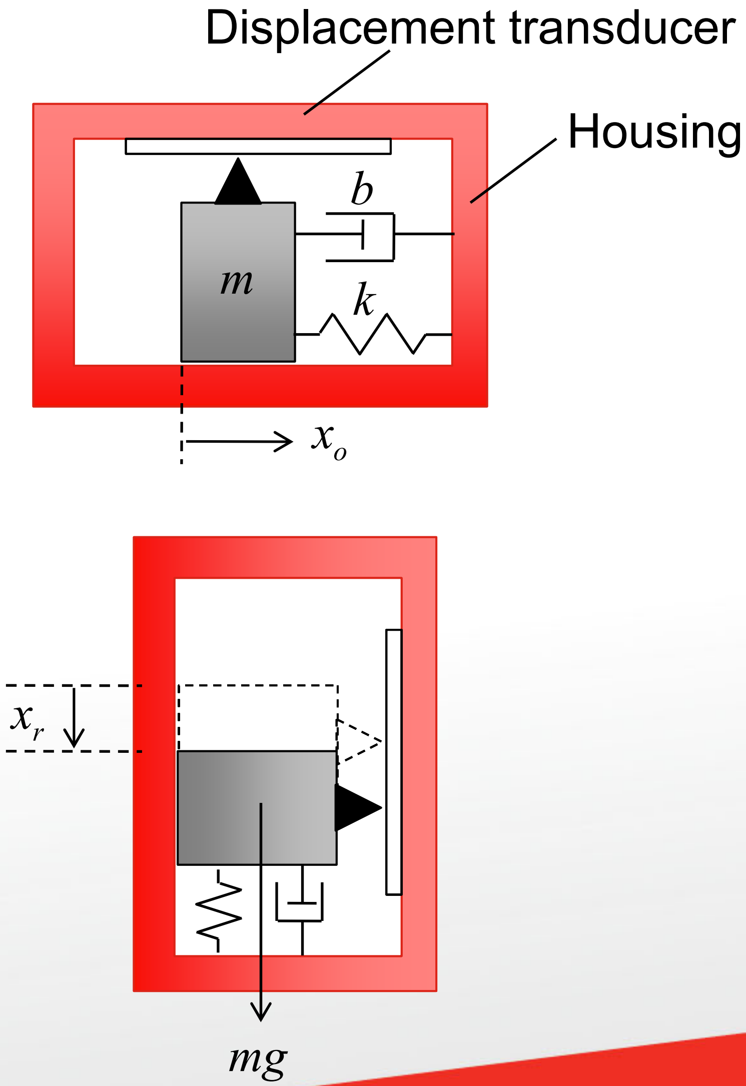
    - Piezoelectric crystal: Deformation results in voltage generation.
    - When housing / object accelerates, the seismic mass will press the piezo-crystal.



- Voltage generated is the measure of acceleration .
- The spring needs to be preloaded to keep mass in contact with the crystal.
- Good for up to 5kHz.

Detect Orientation by Accelerometer

- Accelerometer is used in consumer electronics (e.g. Smartphones) to detect orientation.
- At different orientation, the seismic mass is pulled by gravity differently.
- The mass will move to another position due to the gravitational force.
- This position is measured by the displacement transducer.
- By proper calibration, the orientation can be measured.

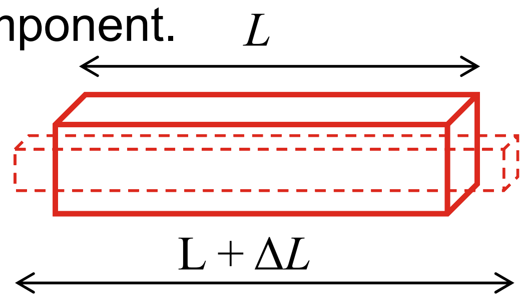


Content

- Position Measurement
    - Measure using myRIO
- Speed Measurement
- Vibration and Acceleration Measurement
- Stress and Strain Measurement
    - Measure using myRIO
- Saving (Sensor) Data using myRIO
- Attachment: Writing FPGA Codes

Stress and Strain Measurement

- To measure the stress and strain in a mechanical component.
  - Strain: Change in length per unit length



$$\varepsilon_{axial} = \frac{\Delta L}{L}$$ Dimensionless

  - Stress: Internal forces.
    - Related to strain: $\sigma_{axial} = E\varepsilon_{axial}$ E: Modulus of Elasticity
- Indirectly measure other quantities such as force and pressure.
- Usage example:
  - Force-controlled gripper, conforming to object shape or applied force.
  - Force-controlled robotic tooling, for polishing of workpiece at precise force.
    - https://www.youtube.com/watch?v=vX_6tOrFmEg
  - Coordinate Measurement Machines (CMM), for measurement of contours. Sensed force allows following of contour.
    - https://www.youtube.com/watch?v=y3WukuT41nc

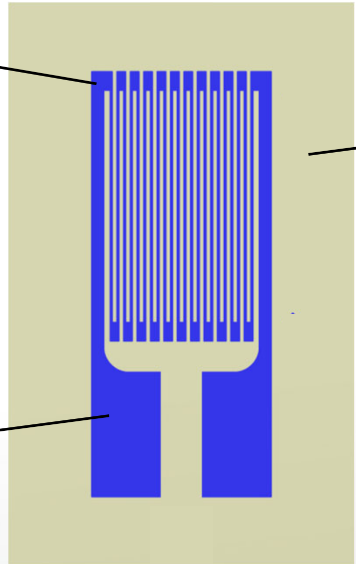
Stress and Strain Measurement (1)

- Electrical Resistance Strain Gauge:

Thin foil of metal, usually constantan, in a grid pattern

Thin plastic backing material, usually polyimide

Metallic pads for soldering

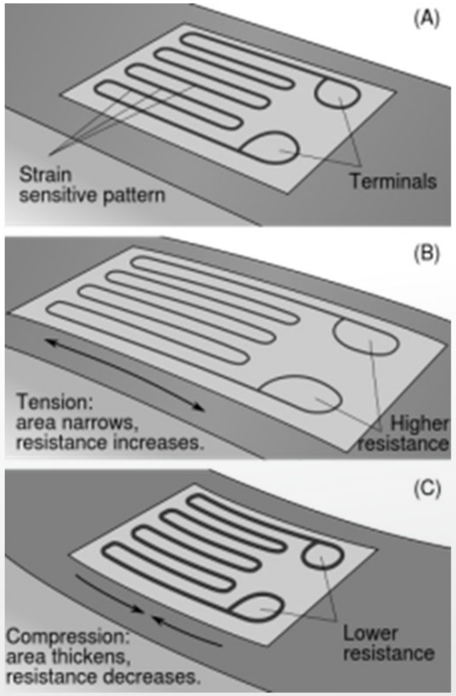


Metal Foil Strain Gauge

https://commons.wikimedia.org/wiki/File:Strain_gauge.svg

How to use Strain Gauge

- Mounting:
  - Adhesively bonded to the surface of the part, whose surface strain is to be measured.
    - Short term measurement: Cyanoacrylate
    - Long term installation: Epoxy
- How does it find out the strain?



- When the component is loaded, the metal foil (which is bonded to the component surface) deforms.
- Resistance changes.
- By measuring the resistance accurately, the strain can be determined.

Deformation of Strain Gauge
https://commons.wikimedia.org/wiki/File:StrainGaugeVisualization.svg

Strain and Resistance

- Strain gauge datasheet: "Gauge Factor" F

$$F = \frac{\frac{\Delta R}{R_0}}{\varepsilon_{\text{axial}}}$$

ΔR : Change in resistance

R_0 : Resistance at the unloaded state

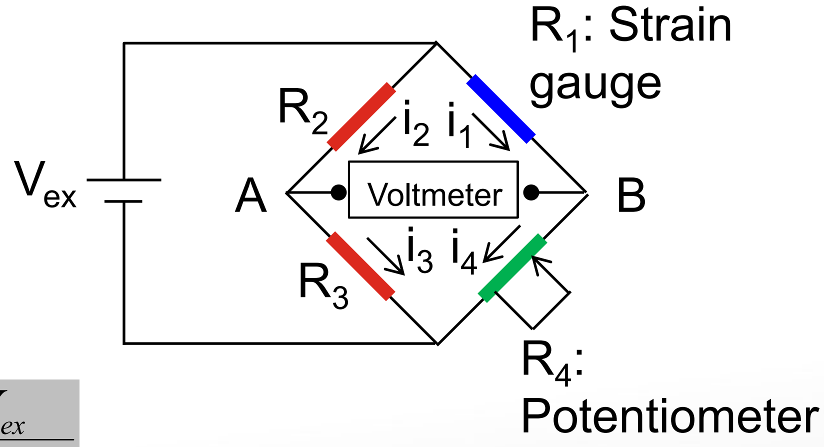
$\varepsilon_{\text{axial}}$: Axial strain

- Calculation example:
  - A 120Ω strain gauge with a gauge factor 2 experiences a change in resistance of 0.024 Ω. How much is the strain?

$$\varepsilon_{\text{axial}} = \frac{\frac{\Delta R}{R_0}}{F} = \frac{\frac{0.024}{120}}{2} = 0.0001$$

Measuring Resistance Changes

- The changes in resistance are usually very small → hard to measure.
- To accurately measure the small changes in resistance, the "Wheatstone Bridge" is commonly used.
  - Static balanced mode:
    - Fixed load is applied.
    - Potentiometer R_4 is tuned until the voltage between A and B is 0.
    - Therefore: $V_A = V_B \Rightarrow i_1 R_1 = i_2 R_2$
    - Also: $i_1 = i_4 = \dfrac{V_{ex}}{R_1 + R_4}$ & $i_2 = i_3 = \dfrac{V_{ex}}{R_2 + R_3}$



- Thus: $i_1 R_1 = i_2 R_2 \rightarrow \dfrac{V_{ex}}{R_1 + R_4} R_1 = \dfrac{V_{ex}}{R_2 + R_3} R_2 \rightarrow \dfrac{R_1 + R_4}{R_1} = \dfrac{R_2 + R_3}{R_2} \rightarrow 1 + \dfrac{R_4}{R_1} = 1 + \dfrac{R_3}{R_2}$

$$\rightarrow \frac{R_1}{R_4} = \frac{R_2}{R_3} \rightarrow R_1 = \frac{R_4 R_2}{R_3}$$

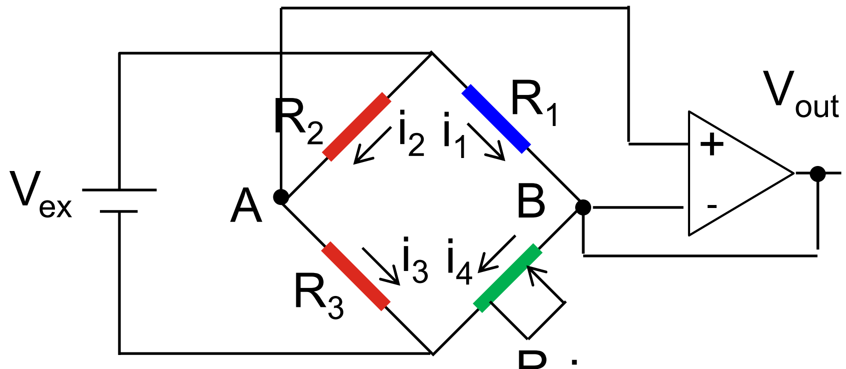
Note: This R_1 is R_0 (from datasheet) + ΔR. We can then calculate ε_{axial} given gauge factor F.

- R_1 accurately measured!

Measuring Resistance Changes

- The static balance mode is for fixed load.
- To measure a varying / dynamic load, we need the...
- dynamic deflection operation:
  - Load is NOT applied first.
  - Potentiometer R_4 is tuned until the voltage between A and B is 0.
  - The changes in the strain gauge resistance R_1 under time-varying load can be determined from changes in output voltage.

*R_1: Strain gauge



R_4: Potentiometer

- The op-amp circuit is differential buffer follower, i.e. $V_{out} = V_+ - V_-$
- Here, $V_{out} = (V_{ex} - i_2R_2) - (V_{ex} - i_1R_1) = i_1R_1 - i_2R_2$
- The excitation voltage is related to the current as: $V_{ex} = i_1(R_1 + R_4) = i_2(R_2 + R_3)$
- (to be continued...)

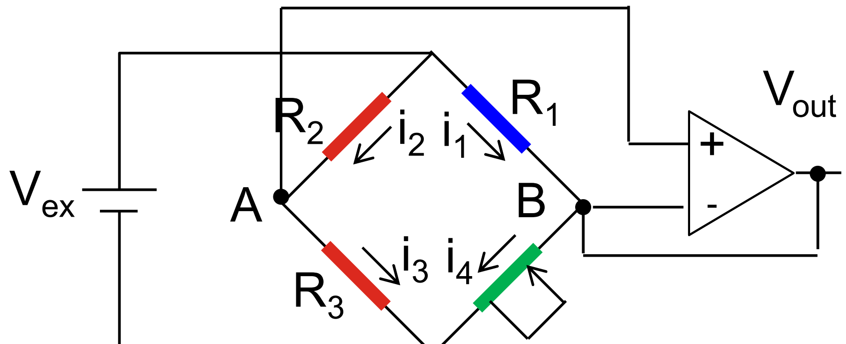
RMIT UNIVERSITY

Measuring Resistance Changes

- (...continued):
  - Solving for i_1 and i_2 in the V_{ex} equation, and substituting into the V_{out} equation, we get:

$$V_{out} = V_{ex}\left(\frac{R_1}{R_1 + R_4} - \frac{R_2}{R_2 + R_3}\right)$$

*R_1: Strain gauge



R4: Potentiometer

  - When the bridge is balanced, V_{out} is zero and R_1 has a known value.
  - When R_1 changes value, the above equation can be used to related the change in R_1 to the change in V_{out}.

$$\frac{V_{out} + \Delta V_{out}}{V_{ex}} = \frac{\Delta V_{out}}{V_{ex}} = \left(\frac{R_1 + \Delta R_1}{R_1 + \Delta R_1 + R_4} - \frac{R_2}{R_2 + R_3}\right)$$

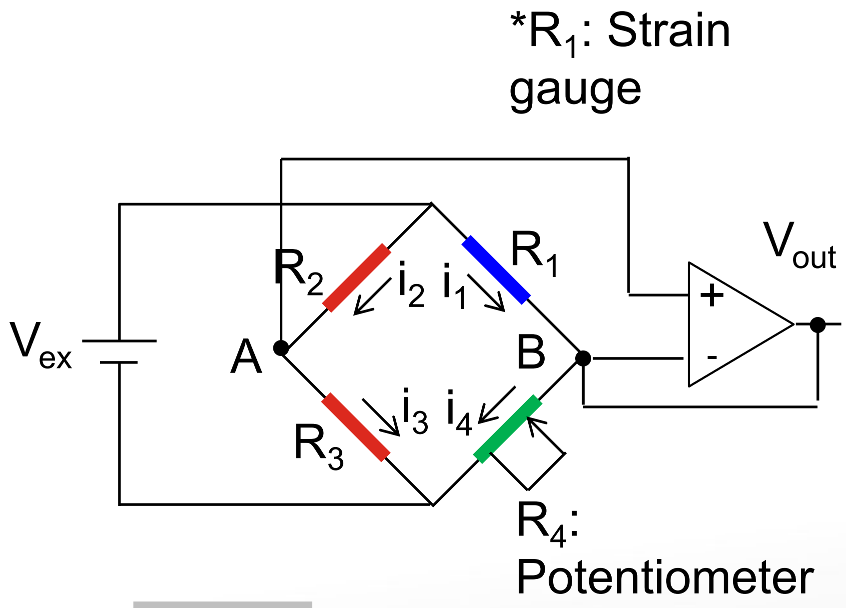
*The first equal sign is because V_{out} = 0

  - (to be continued...)

Measuring Resistance Changes

- (...continued):
  - After rearranging:

$$\frac{\Delta R_1}{R_1} = \frac{\frac{R_4}{R_1}\left(\frac{\Delta V_{out}}{V_{ex}} + \frac{R_2}{R_2 + R_3}\right)}{\left(1 - \frac{\Delta V_{out}}{V_{ex}} - \frac{R_2}{R_2 + R_3}\right)} - 1$$



  - Thus: measure the change in the output voltage ΔV_{out}.
  - Then: ΔR_1 can be determined.
  - Finally, the strain can be determined from: $F = \frac{\frac{\Delta R_1}{R_1}}{\varepsilon_{\text{axial}}}$
  - If we assume $R_2 = R_3$, then R1 = R4 at balanced stage. It can be shown that:

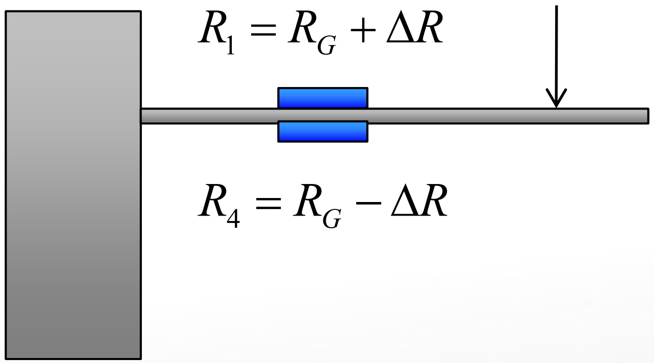
$$\frac{\Delta V_{out}}{V_{ex}} = \frac{F \cdot \varepsilon_{axial}}{4} \cdot \frac{1}{\left(1 + \frac{F \cdot \varepsilon_{axial}}{2}\right)}$$

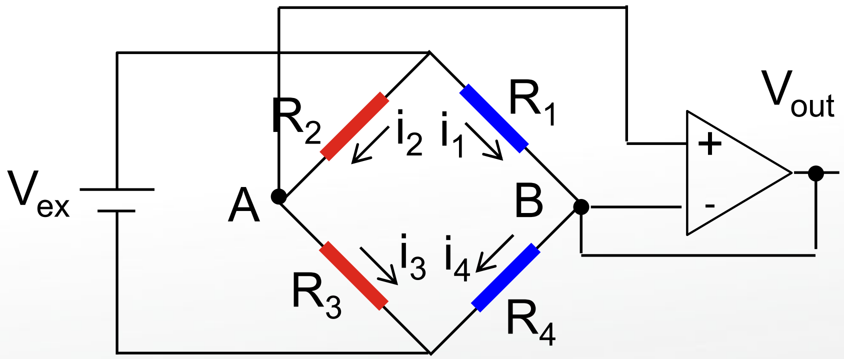
Nonlinear

RMIT UNIVERSITY

Increase Sensitivity

- The sensitivity of the bridge can be doubled by making both gauges active, but in different directions.
- For e.g. In a bending beam application,
  - One gauge (R_1) is mounted in tension
  - Another gauge (R_4) is mounted in compression.

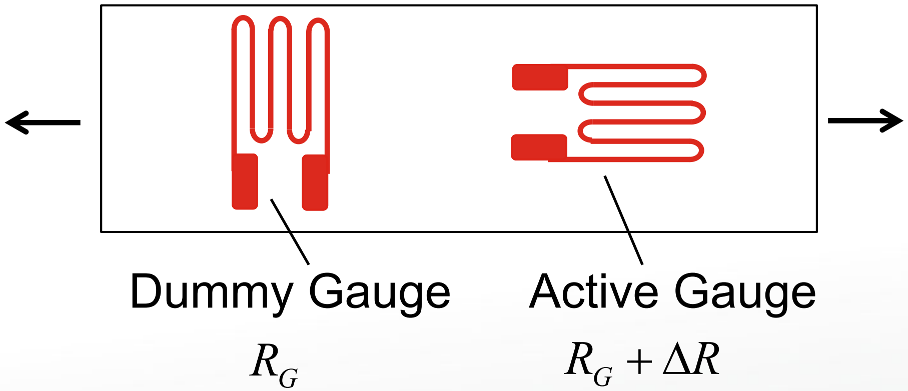


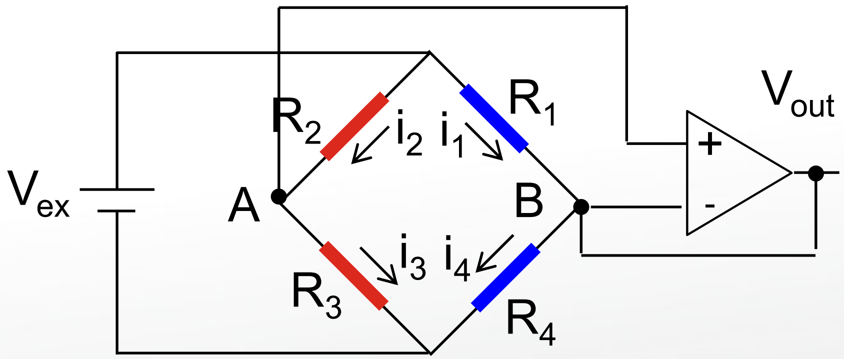


  - The output voltage is linear and approximately doubles the output of quarter-bridge circuit. $\frac{V_{out}}{V_{ex}} = \frac{F \cdot \varepsilon_{axial}}{2}$

Temperature Compensation

- Temperature can affect the resistance of the strain gauge, which would lead to wrong measurement.
- To reduce the effect of temperature on the strain measurement:
  - Use two strain gauges in the bridge.
    - An active gauge (R_1) in the direction of strain.
    - A dummy gauge (R_4) one perpendicular to the direction of strain.



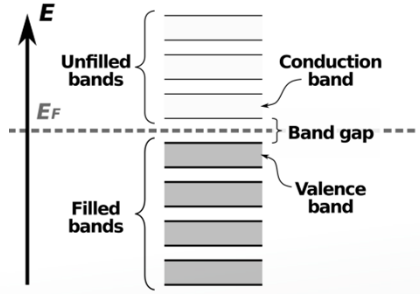


  - Temperature changes are identifical in the two gauges, thus the ratio of their resistance does not change.
  - Therefore the voltage V_{out} does not change due to temperature.

RMIT UNIVERSITY

Piezoresistive Strain Gauge

- Apart from metal foil strain gauges, there are also strain gauges made of semiconductor material.
- They take advantage of the piezoresistive effect:
  - Change in resistance when mechanical strain is applied.
- The change in resistance is greater than what can be explained by simple geometric deformation.
  - On the atomic scale, the change in volume causes the energy gap between the valence and conduction bands to change.
  - Therefore it becomes harder for the electrons to be raised into the conduction band.

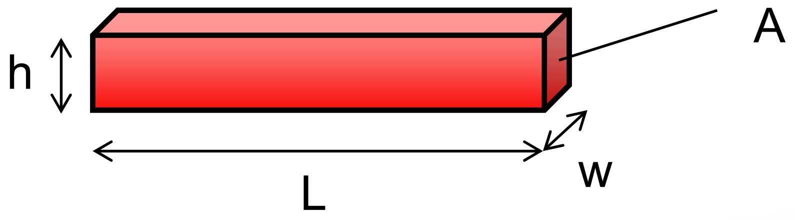


Semiconductor Band Structure
https://commons.wikimedia.org/wiki/File:Semiconductor_band_structure_(lots_of_bands_2).svg

- The change in resistance due to piezoresistance is approximately two orders of magnitude higher than in non-piezoresistive materials.
  - Therefore, semiconductor-based strain gauges are much more sensitive to deformation.

Derivation of Gauge Factor

- Previously, we saw that the gauge factor F is: $F = \dfrac{\frac{\Delta R}{R_0}}{\varepsilon_{\text{axial}}}$
- Let's derive this together.
- We approximate the metal foil grid lines as a single rectangular conductor.
- The total resistance is: $R = \dfrac{\rho L}{A}$



- Where ρ is the resistivity:
  - Resistance if L = 1m and A = 1m^2
- Then: $dR = \dfrac{L}{A} d\rho + \dfrac{\rho}{A} dL - \dfrac{\rho L}{A^2} dA$ ⇒ $\dfrac{dR}{R} = \dfrac{d\rho}{\rho} + \dfrac{dL}{L} - \dfrac{dA}{A}$
- Because the cross-sectional area is: $A = wh$
- Therefore: $\dfrac{dA}{A} = \dfrac{w \cdot dh}{wh} + \dfrac{h \cdot dw}{wh} = \dfrac{dh}{h} + \dfrac{dw}{w}$

RMIT UNIVERSITY

Derivation of Gauge Factor

- From Poisson's ratio (ratio of transverse and axial strain):

$$\frac{dh}{h} = -\upsilon \frac{dL}{L} \qquad \frac{dw}{w} = -\upsilon \frac{dL}{L}$$

- Therefore: $$\frac{dA}{A} = -2\upsilon \frac{dL}{L} = -2\upsilon \varepsilon_{\text{axial}}$$

- Substituting back into $\frac{dR}{R} = \frac{d\rho}{\rho} + \frac{dL}{L} - \frac{dA}{A}$ gives: $\frac{dR}{R} = \frac{d\rho}{\rho} + (1+2\upsilon)\varepsilon_{\text{axial}}$

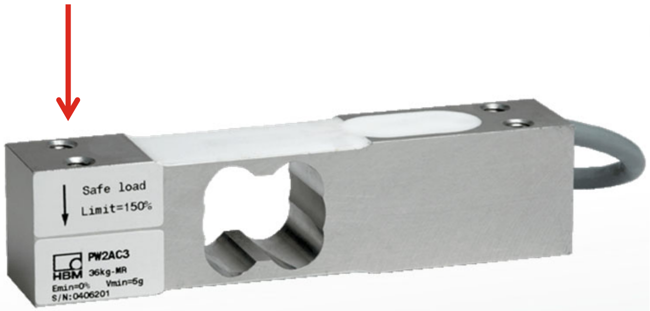
- Dividing through $\varepsilon_{\text{axial}}$ gives:

$$\underbrace{\frac{dR/R}{\varepsilon_{\text{axial}}}}_{F} = \underbrace{\frac{d\rho/\rho}{\varepsilon_{\text{axial}}}}_{\text{Piezoresistive Effect}} + \underbrace{(1+2\upsilon)}_{\text{due to change in length and area}}$$

Force Measurement

- Load cell is a sensor used to measure force.
- It contains an internal flexure element, usually with several strain gauges mounted on its surface.
- The flexural element's shape is designed so that the strain gauge outputs can be easily related to the applied force.

Compression Load Cell
https://commons.wikimedia.org/wiki/File:Silomer_100kN.jpg



Double Beam Load Cell
https://commons.wikimedia.org/wiki/File:DoubleBBeam_WZ.png

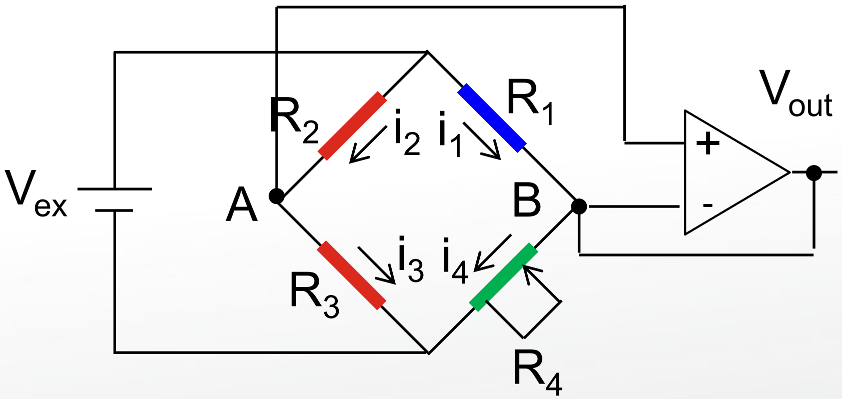
S Load Cell
https://en.wikipedia.org/wiki/File:S_Load_Cell.PNG

Reading & Calibrating Force Sensor

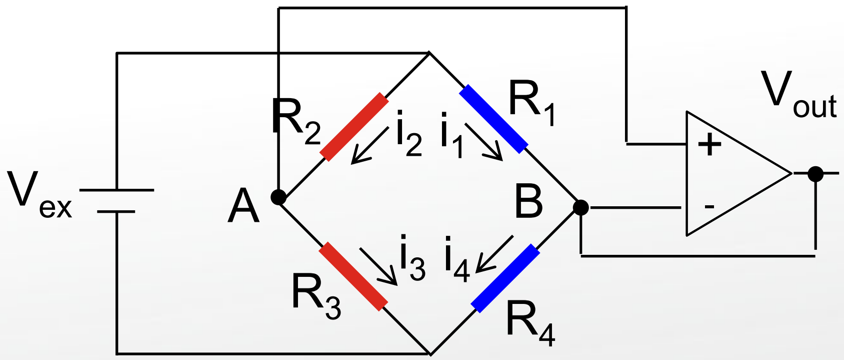
- Just now, we also introduced the quarter Wheatstone bridge, and then the half Wheatstone bridge.
  - It was mentioned that the half bridge would have twice the sensitivity of the quarter bridge.

$$\frac{\Delta V_{out}}{V_{ex}} = \frac{F \cdot \varepsilon_{axial}}{4} \cdot \frac{1}{\left(1 + \frac{F \cdot \varepsilon_{axial}}{2}\right)}$$

$$\frac{\Delta V_{out}}{V_{ex}} = \frac{F \cdot \varepsilon_{axial}}{2}$$



*R1: Strain gauge
R4: Potentiometer

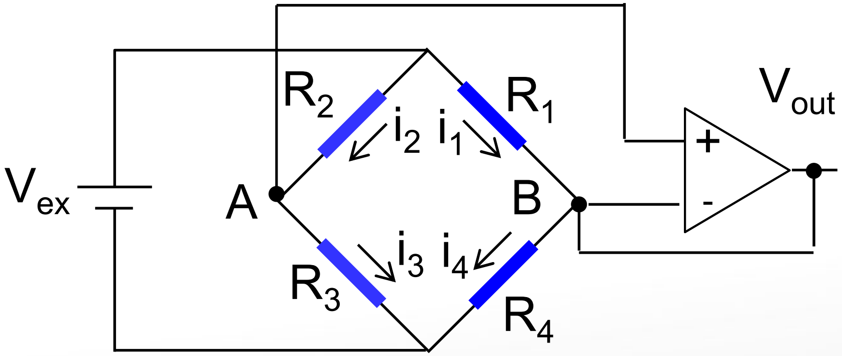


*R1 & R4: Strain gauges

Reading & Calibrating Force Sensor

- By using four strain gauges, we have a "Full bridge" and it doubles the sensitivity of the half bridge".

$$\frac{\Delta V_{out}}{V_{ex}} = F \cdot \varepsilon_{axial}$$

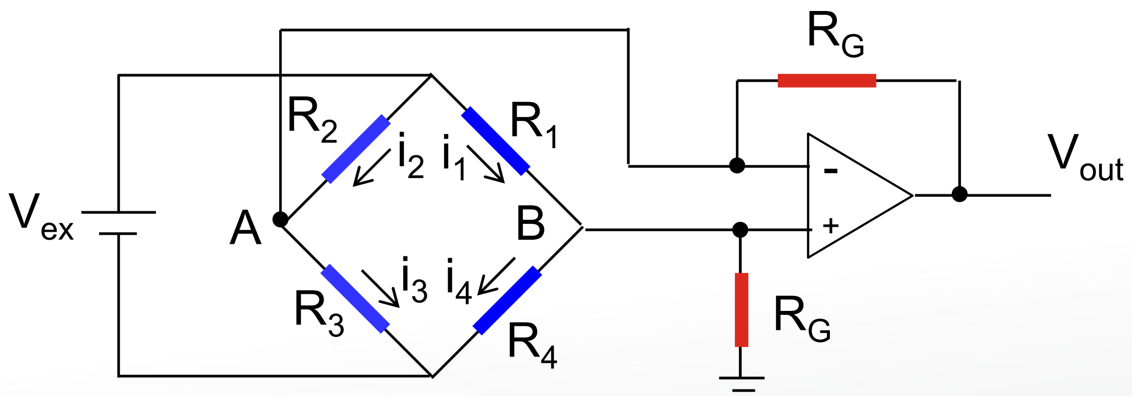


ALL: Strain gauges

- Most load cells (including the one provided in project i.e. TAS606) has this full bridge built into it.

Reading & Calibrating Force Sensor

- Nevertheless, even with full bridge circuit, the potential difference between A and B could still be too small.
- Therefore we need to amplify it.
- Use the following differential amplifier circuit:

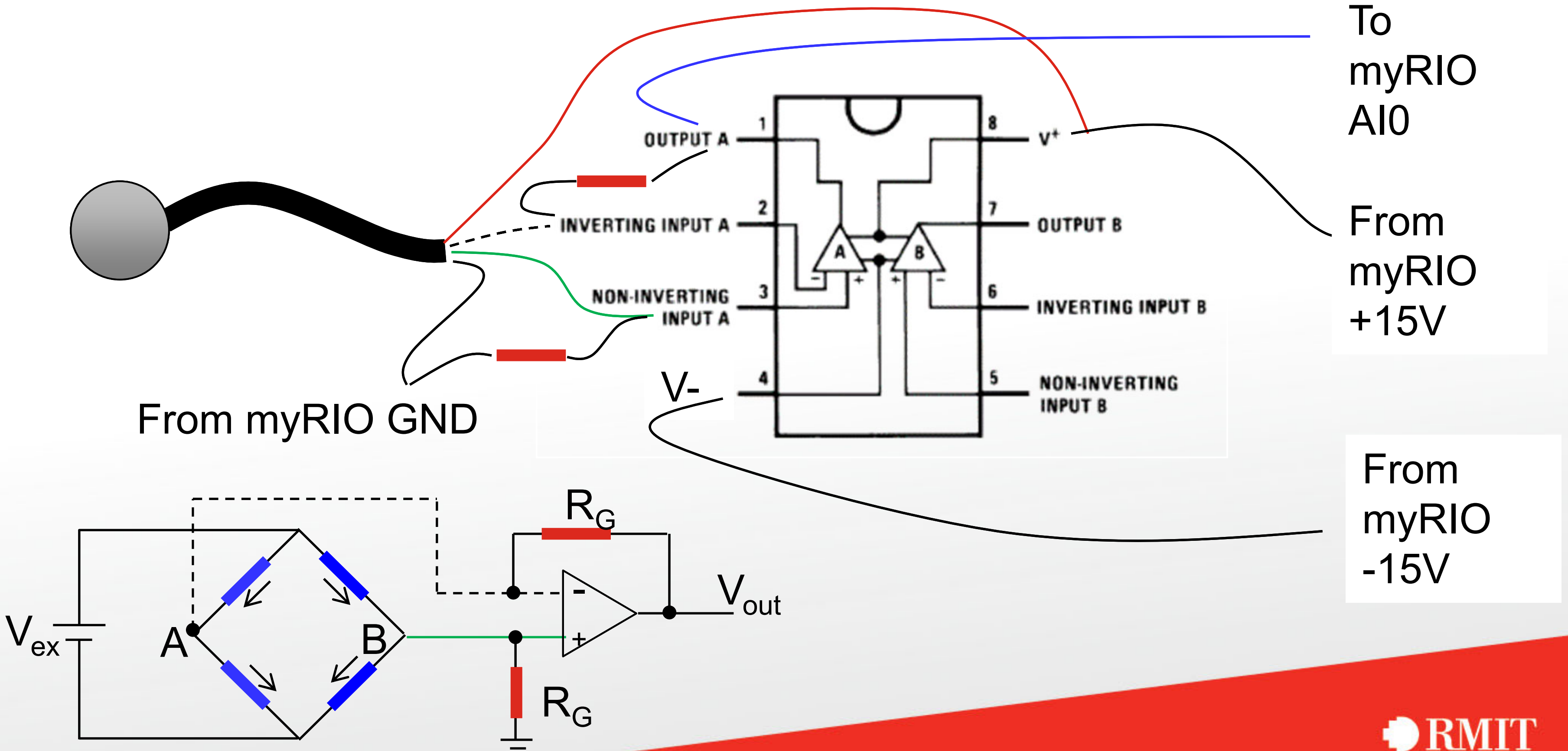


ALL: Strain gauges

- Assume at unloaded state, all four strain gauges have the resistance R.
- Then: $V_{out} = \frac{R_G}{0.5R} V_{AB}$
- Choose appropriate R_G to get high enough gains.

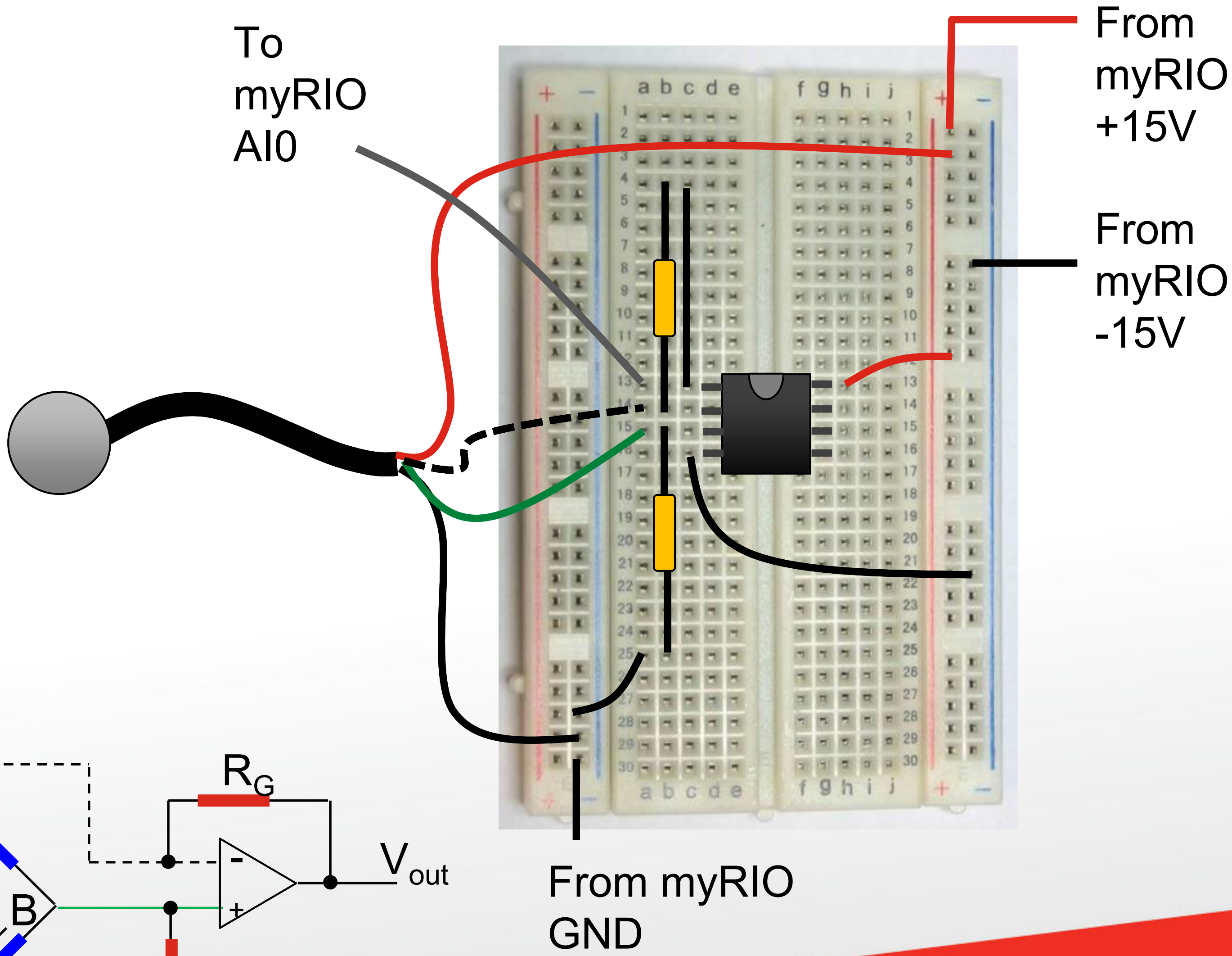
Reading & Calibrating Force Sensor

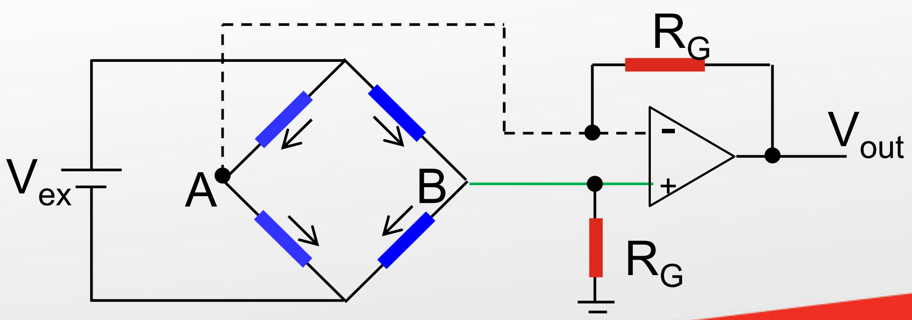
- For TAS606, R has been measured to be around 360 Ohm.
- R_G is chosen to be 1M Ohm, so the gain is 5555.
- The operational amplifier is an LM358, and is connected as shown:



Reading & Calibrating Force Sensor

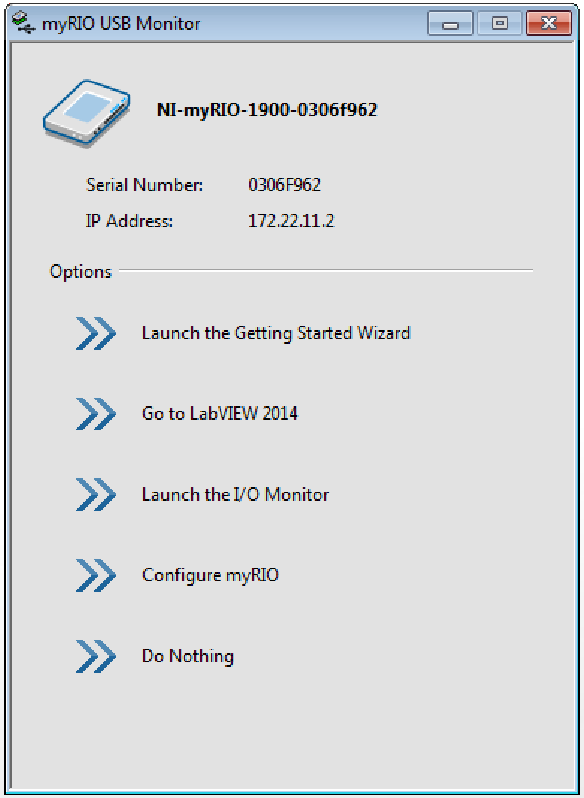
- More details:





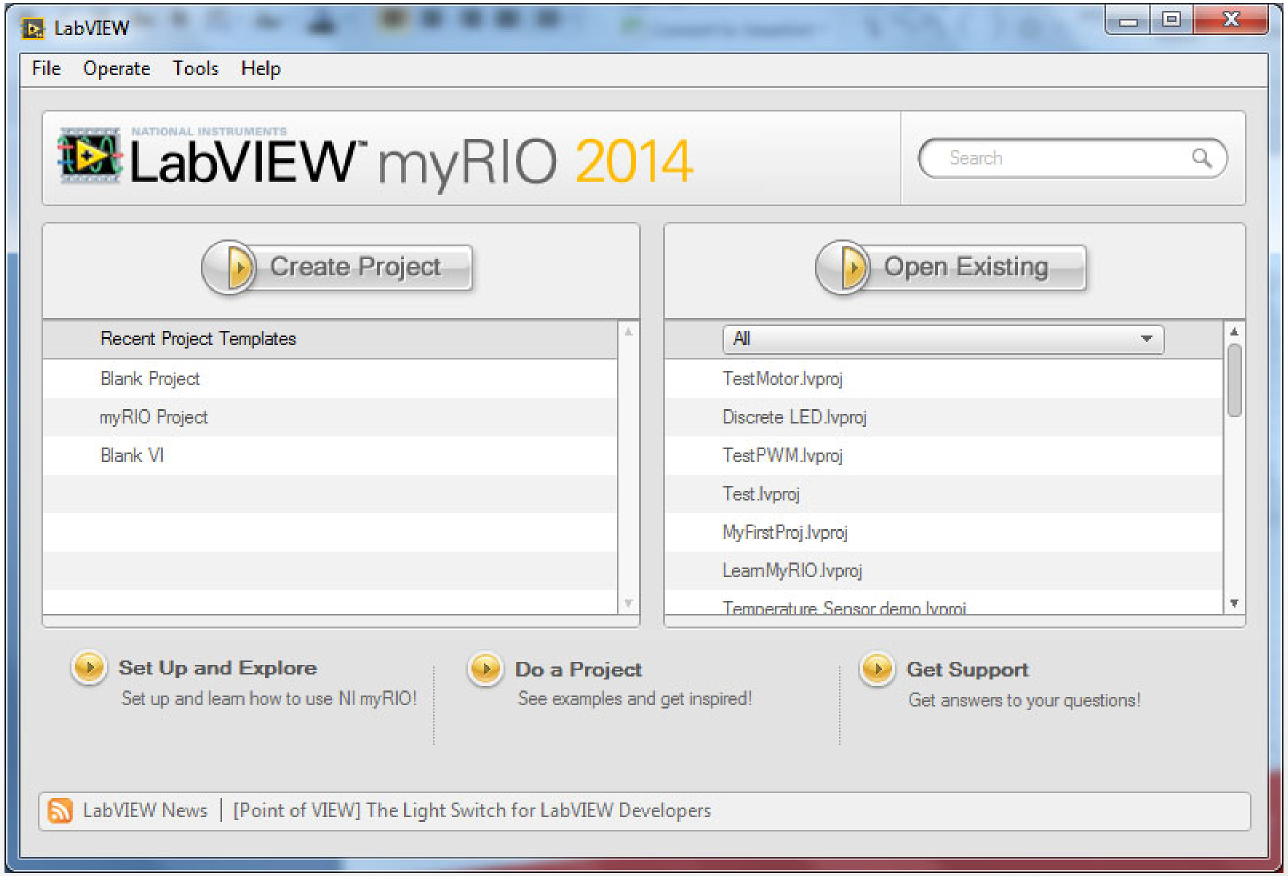
Setting up the Project

- Now let's set up a project in LabVIEW to read in the force sensor signals.
- As practice, let's start all over again.
- Unplug and plug-in myRIO device again.
- On the pop-up menu, choose
  "Go to LabVIEW 201x".



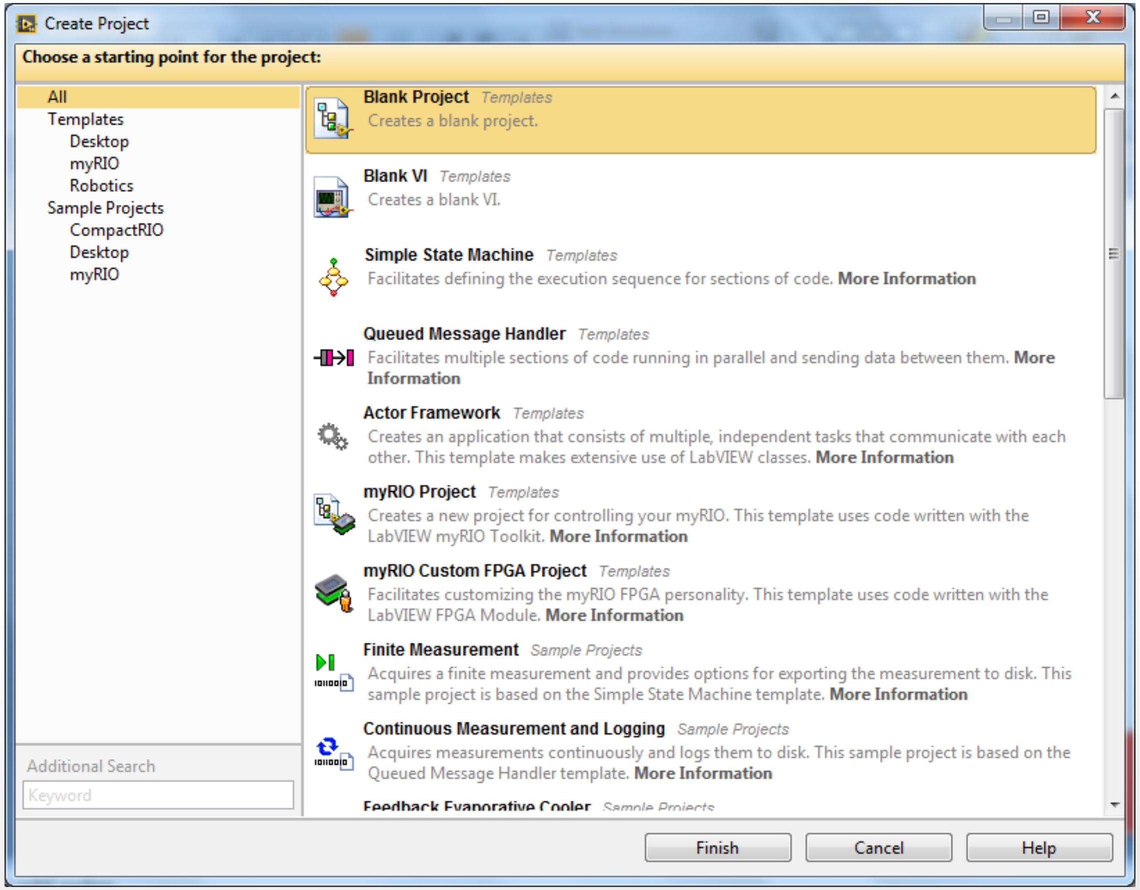
Setting up the Project

- Click on “Create Project”



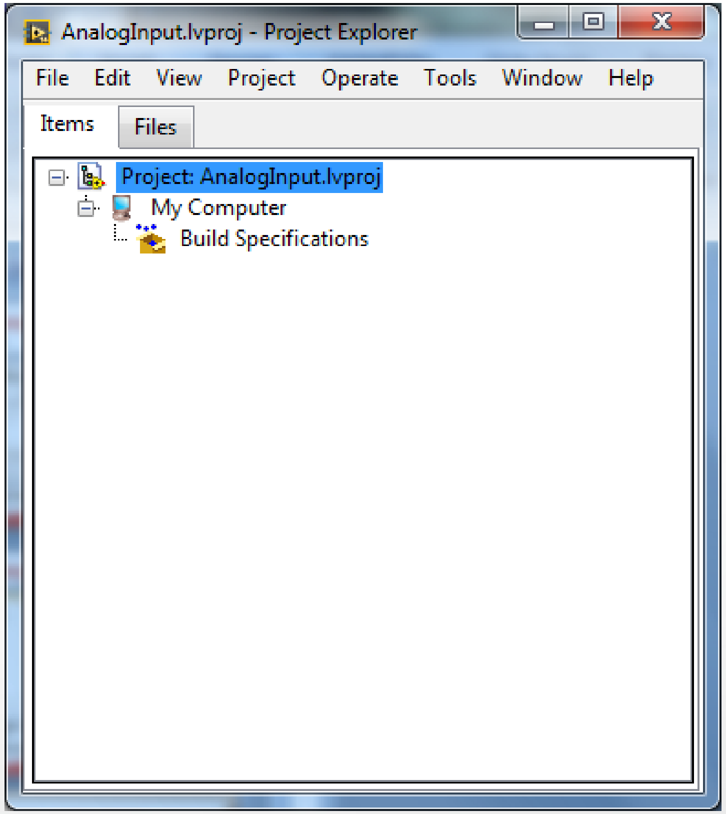
Setting up the Project

- Choose blank project.



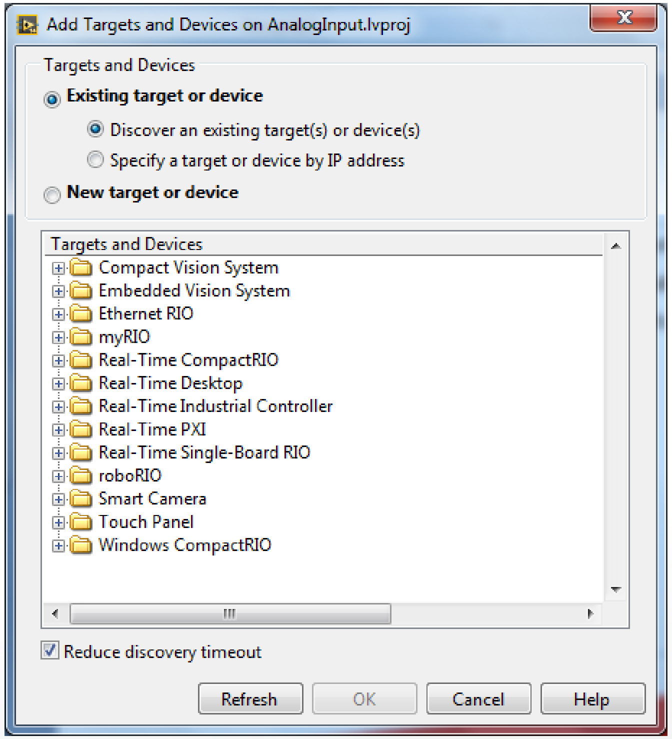
Setting up the Project

- You will see the project window.
- Right click on “Project: Untitled Project 1” → Save as → “AnalogInput”.



Setting up the Project

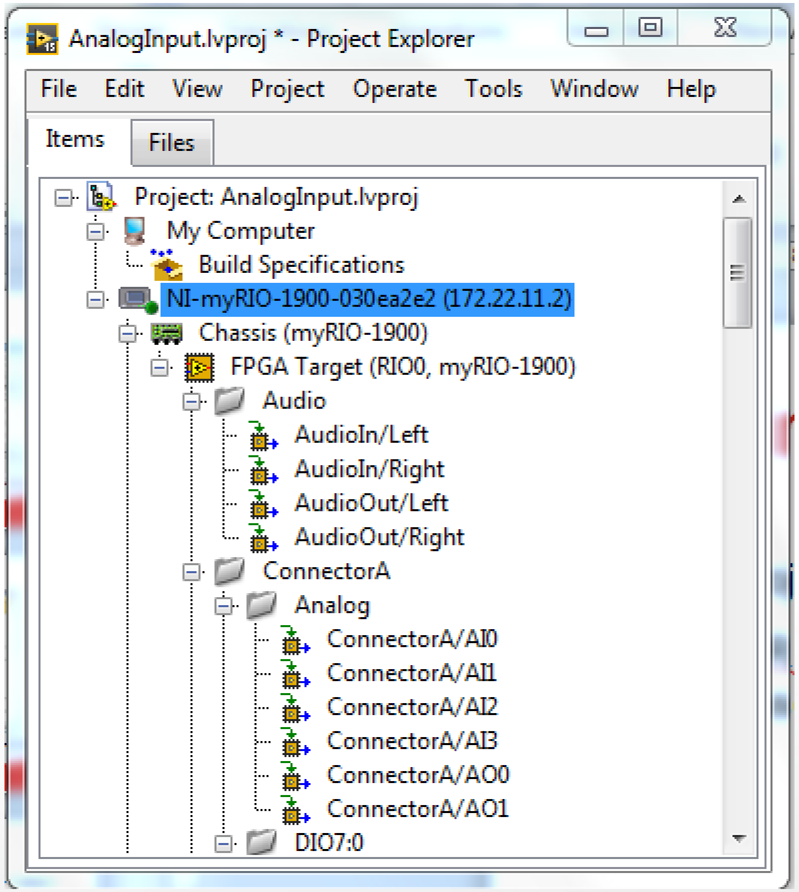
- Right click on Project AnalogInput → New → Targets and Devices



- Existing target or device:
  - Discover an existing target or device.
  - Or specify a target or device by IP address. (myRIO's IP is 172.22.11.2)
- Click the "+" sign before myRIO.

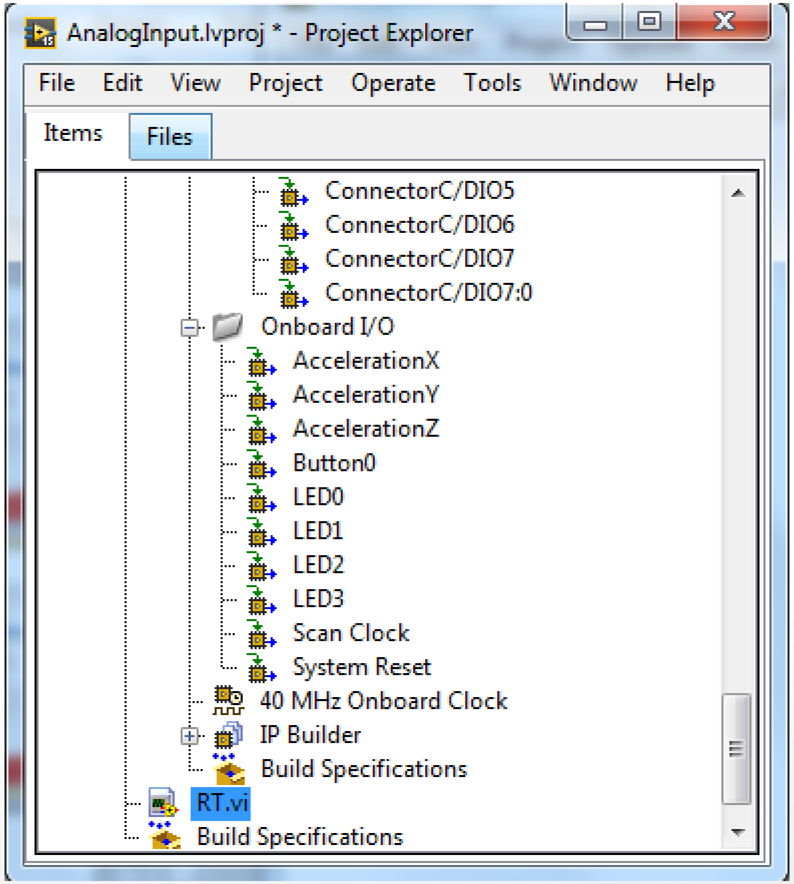
Setting up the Project

- The project tree now looks like this:



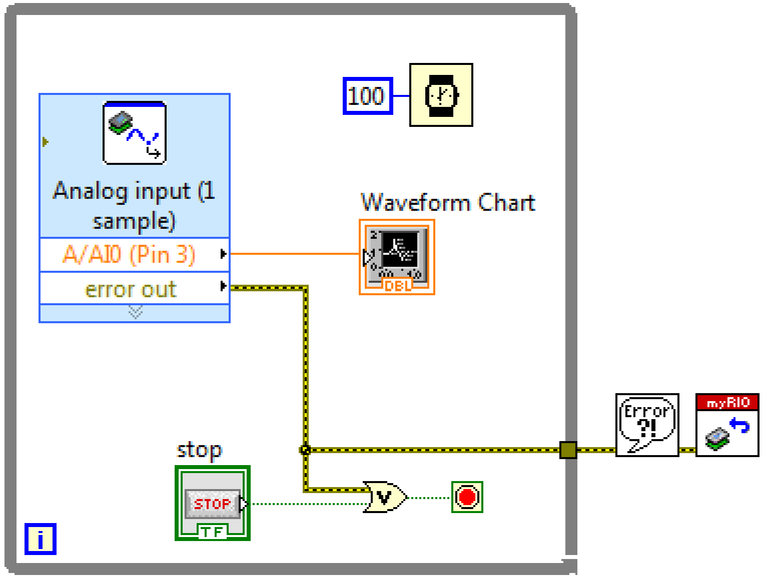
Setting up the Project

- Right click on "NI-myRIO-1900…" → New → VI
- A VI will open.
- Save it as RT.vi.
- Now we are ready to program the VI for reading analog inputs.

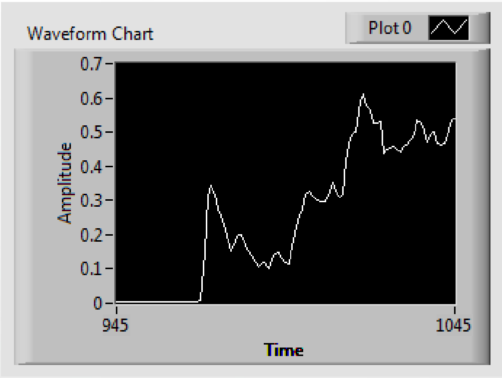


Reading & Calibrating Force Sensor

- Next, create the following VI in RT.vi:



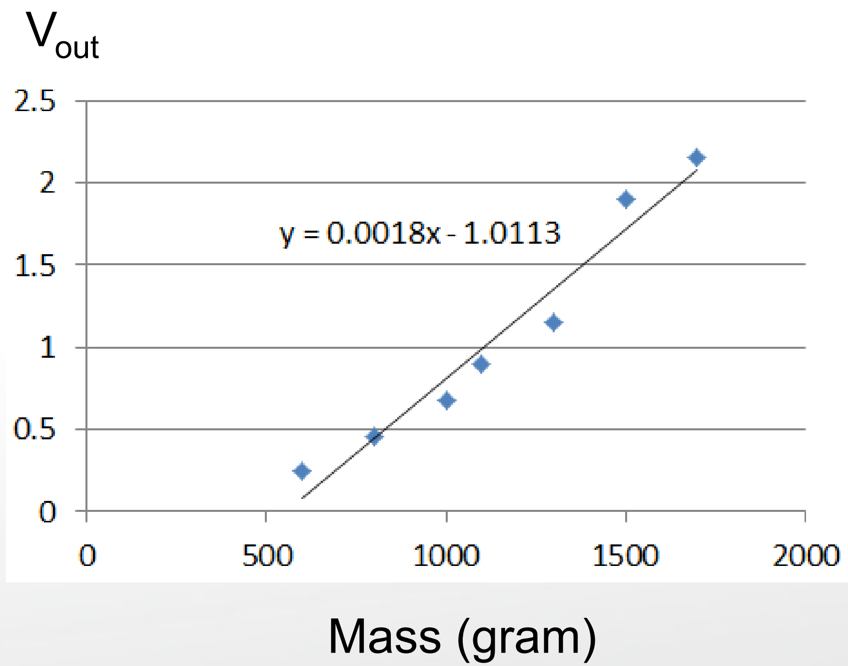
- Press the force sensor and you should be able to see the values in the waveform chart varying.



Reading & Calibrating Force Sensor

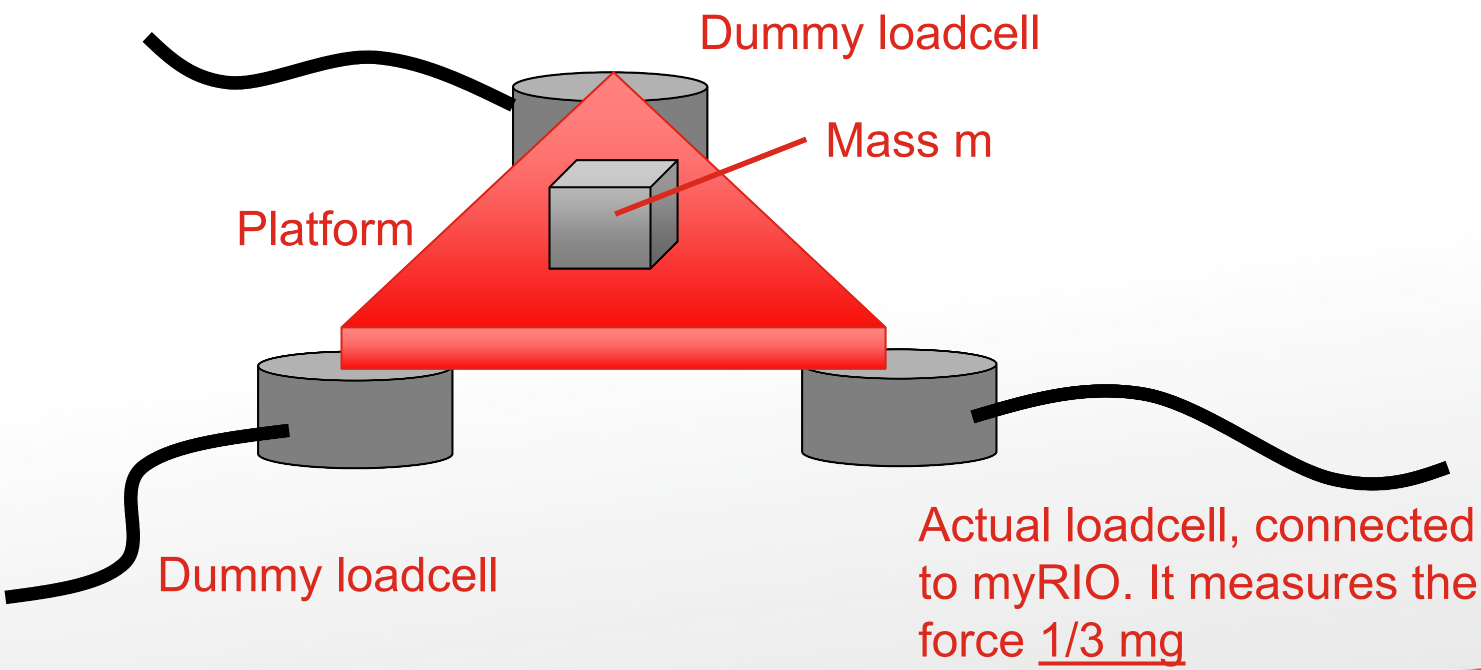
- We may be able to calculate the theoretical values of V_{out}, but it is easier and more accurate to do our own calibration.
- Put objects of known mass on top of the force sensor and note down the value. E.g.

Mass (gram)	V_{out} (V)
600	0.25
800	0.45
1000	0.675
1100	0.9
1300	1.15
1500	1.9
1700	2.15



Reading & Calibrating Force Sensor

- It is hard to put objects on top of 1 load cell.
- So you may consider doing this:



Reading & Calibrating Force Sensor

- From the best fit curve, we can calculate the force applied onto the force sensor, based on the measured V_{out}:

$$V_{out} = 0.0018 \times m - 1.0113$$

$$m = \frac{(V_{out} + 1.0113)}{0.0018} \text{ in gram}$$

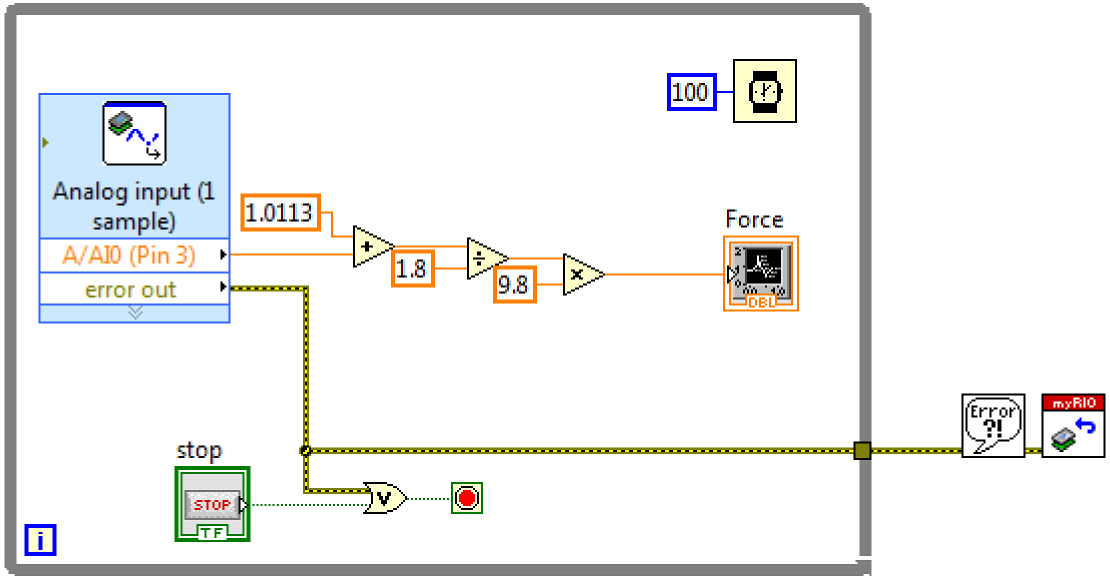
$$= \frac{(V_{out} + 1.0113)}{0.0018 \times 1000} \text{ in kilogram}$$

$$F = mg$$

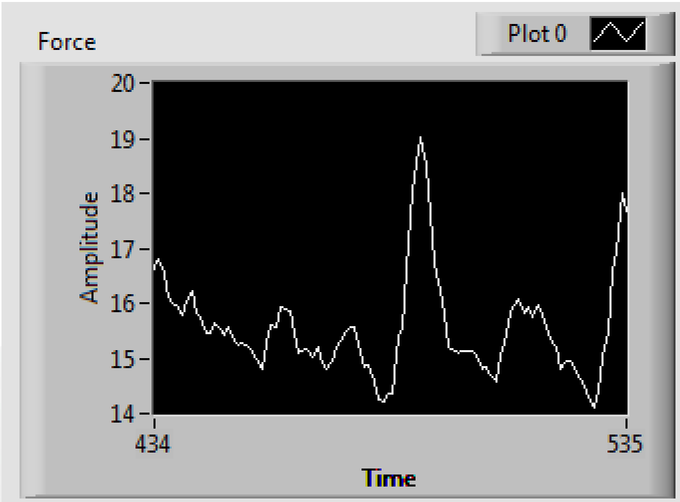
$$= \frac{(V_{out} + 1.0113)}{0.0018 \times 1000} \times 9.8$$

Reading & Calibrating Force Sensor

- We can therefore update our RT.vi as follows:



Note: All the sensors + Op-Amp combinations are different. Do not use the values on this slide for your project!



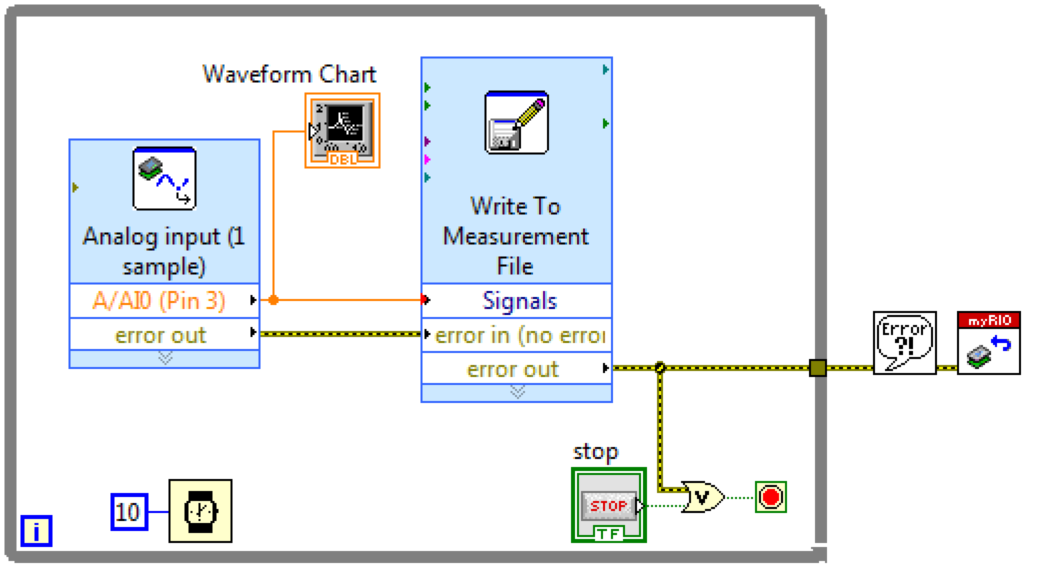
- Run the code, and you will see the force value changes when you press the force sensor.

Content

- Position Measurement
    - Measure using myRIO
- Speed Measurement
- Vibration and Acceleration Measurement
- Stress and Strain Measurement
    - Measure using myRIO
- Saving (Sensor) Data using myRIO
- Attachment: Writing FPGA Codes

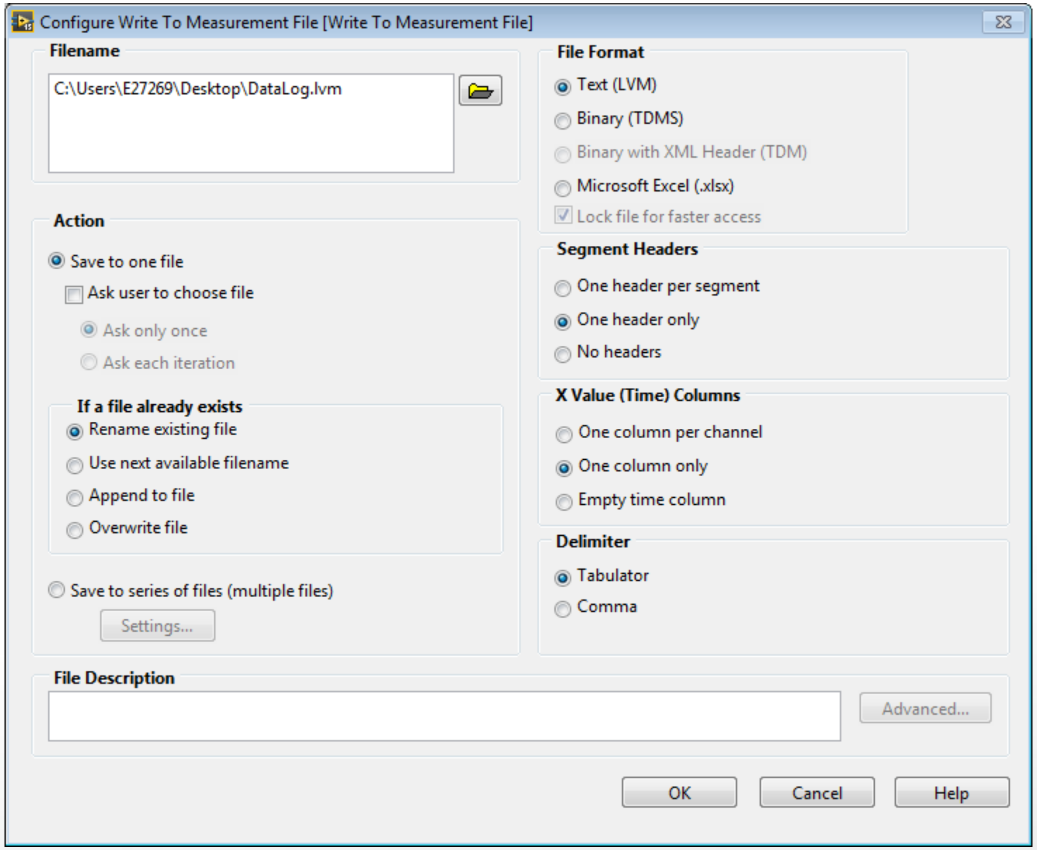
Saving Measurement Data in myRIO

- Let's use the AnalogInput project again, but now with logging of data.



Saving Measurement Data in myRIO

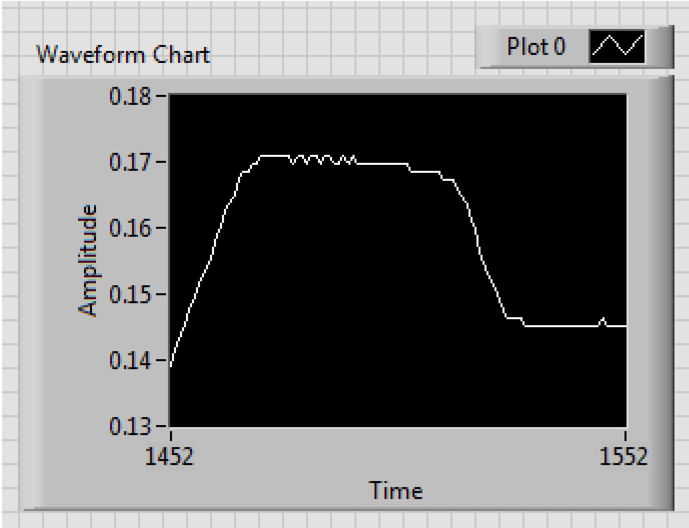
- The Write the Measurement File express VI has been configured as follows:



- Note: In this example, data is to be saved onto desktop.

Saving Measurement Data in myRIO

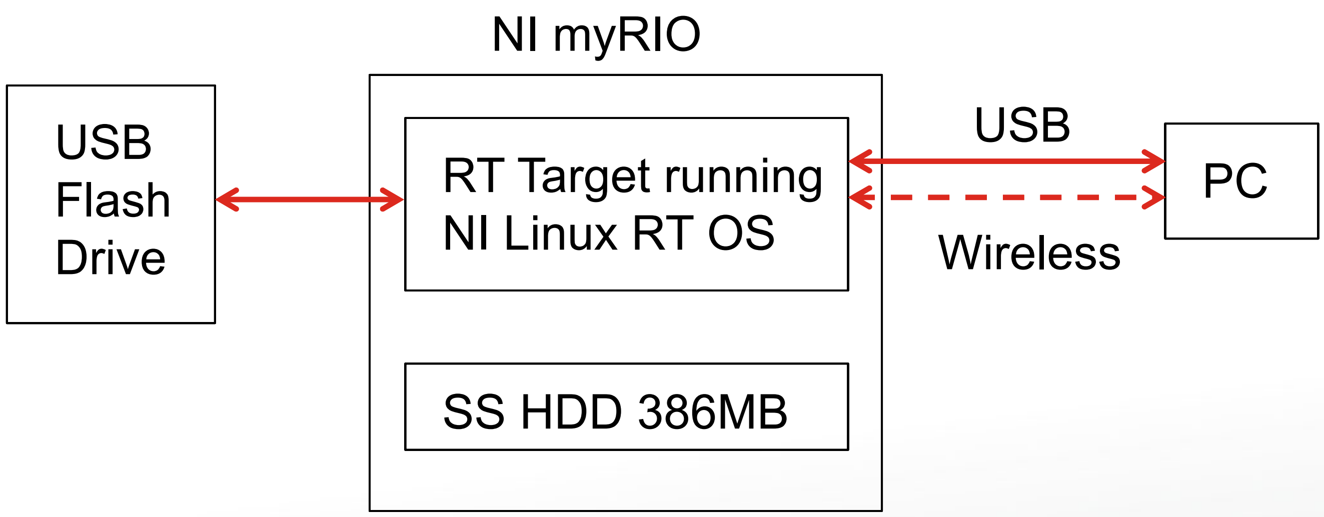
- Now, run the VI and turn the potentiometer.



- After a while, press the stop button.
- Now, try to look for the measurement file on the desktop.
  - It's not in the designated folder!
  - Where is it?

Saving Measurement Data in myRIO

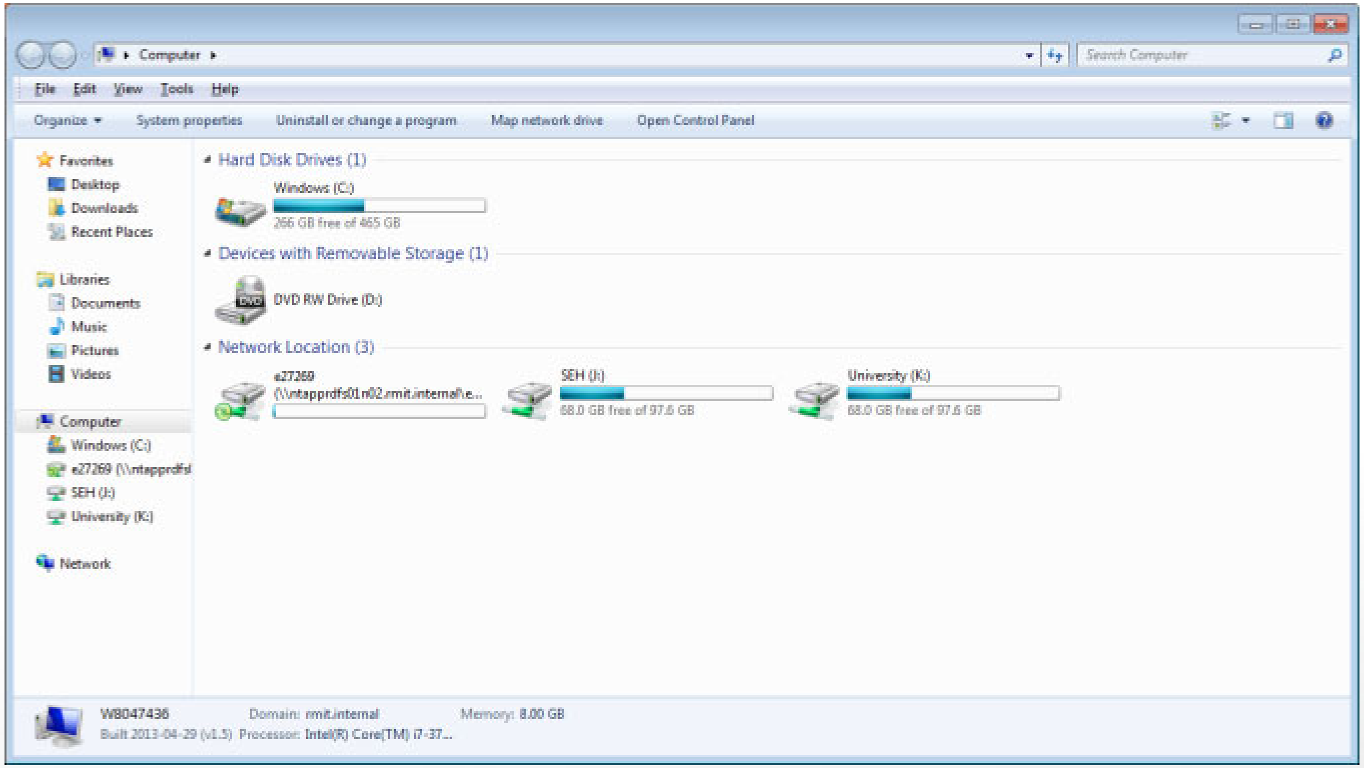
- The following diagram shows the NI myRIO file system:



- The data is actually saved in the Solid State Hard Disk Drive of myRIO.

Saving Measurement Data in myRIO

- To access the data, go to Computer.

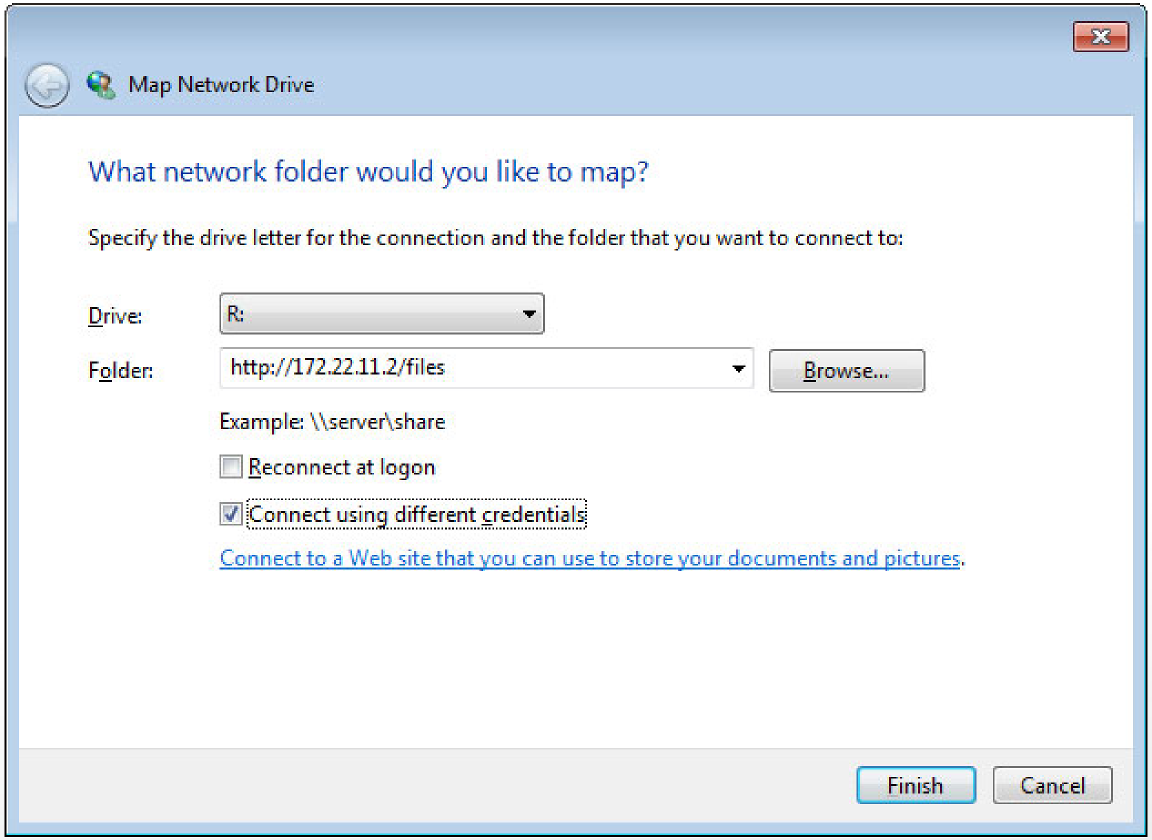


- Click “Map network drive”.

RMIT UNIVERSITY

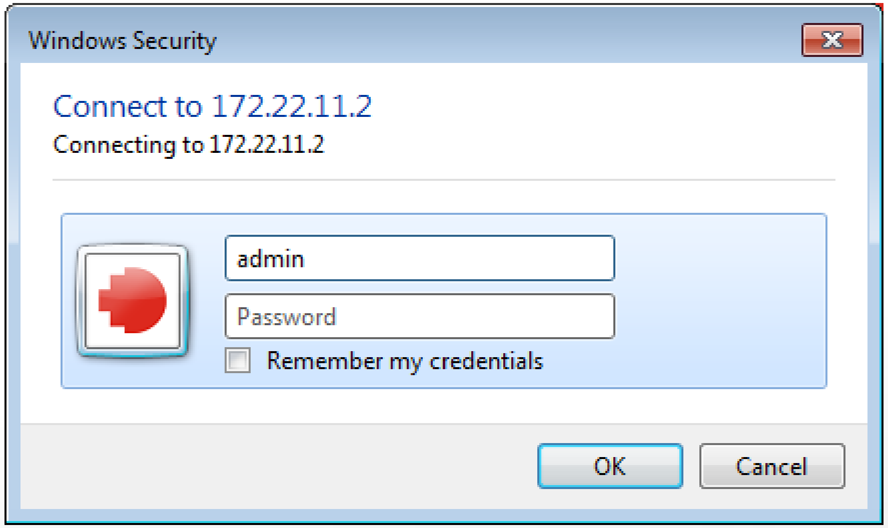
Saving Measurement Data in myRIO

- Choose any available drive (e.g. R for myRIO).
- Key in the folder as shown.
- Uncheck “Reconnect at logon”.
- Check “Connect using different credentials”.
- Click Finish



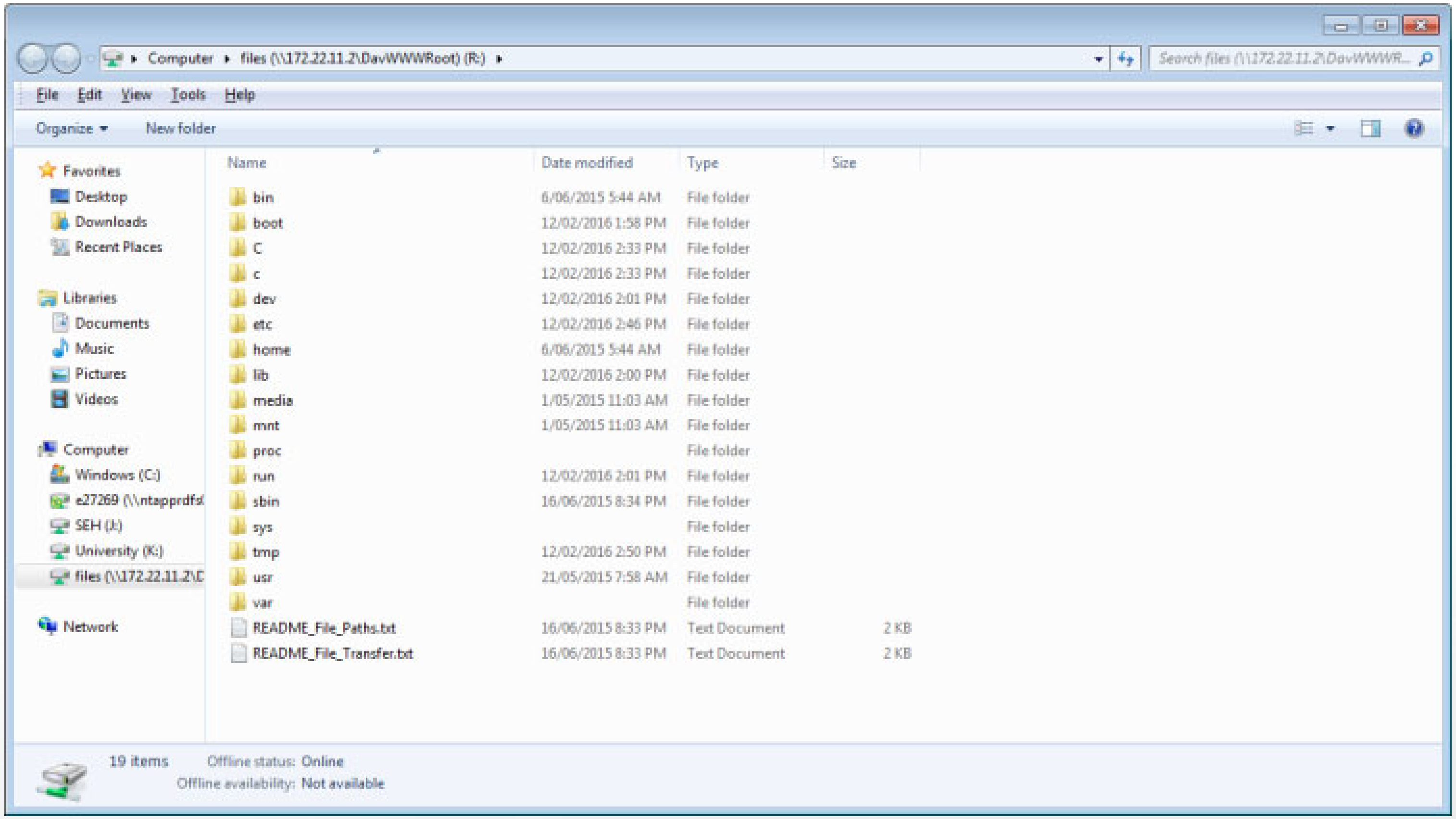
Saving Measurement Data in myRIO

- A Windows Security dialog will pop up.
- Key in username as “admin”.
- Password is empty.



Saving Measurement Data in myRIO

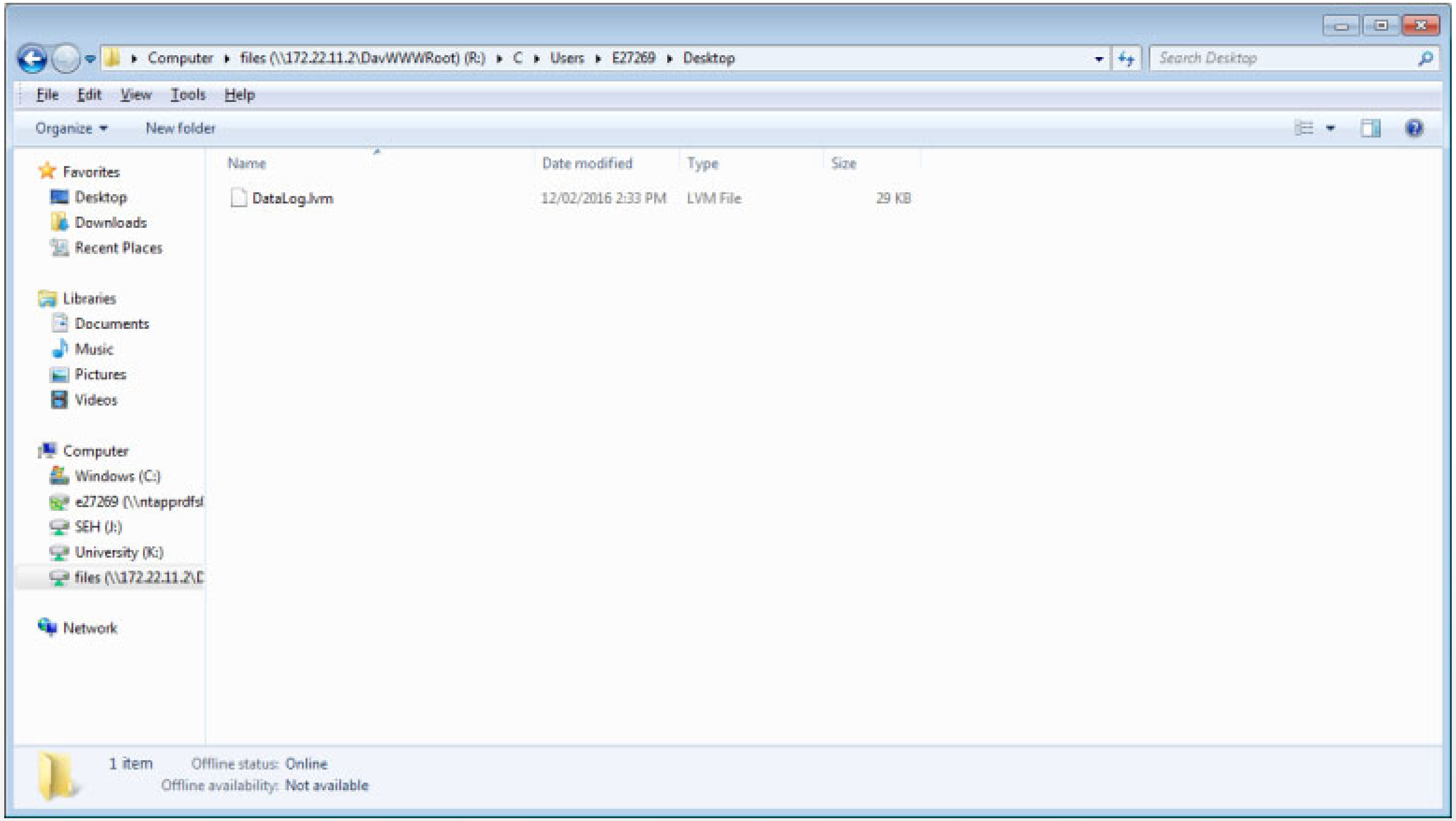
- Now the folder appears and you can look for the measurement file.



RMIT UNIVERSITY

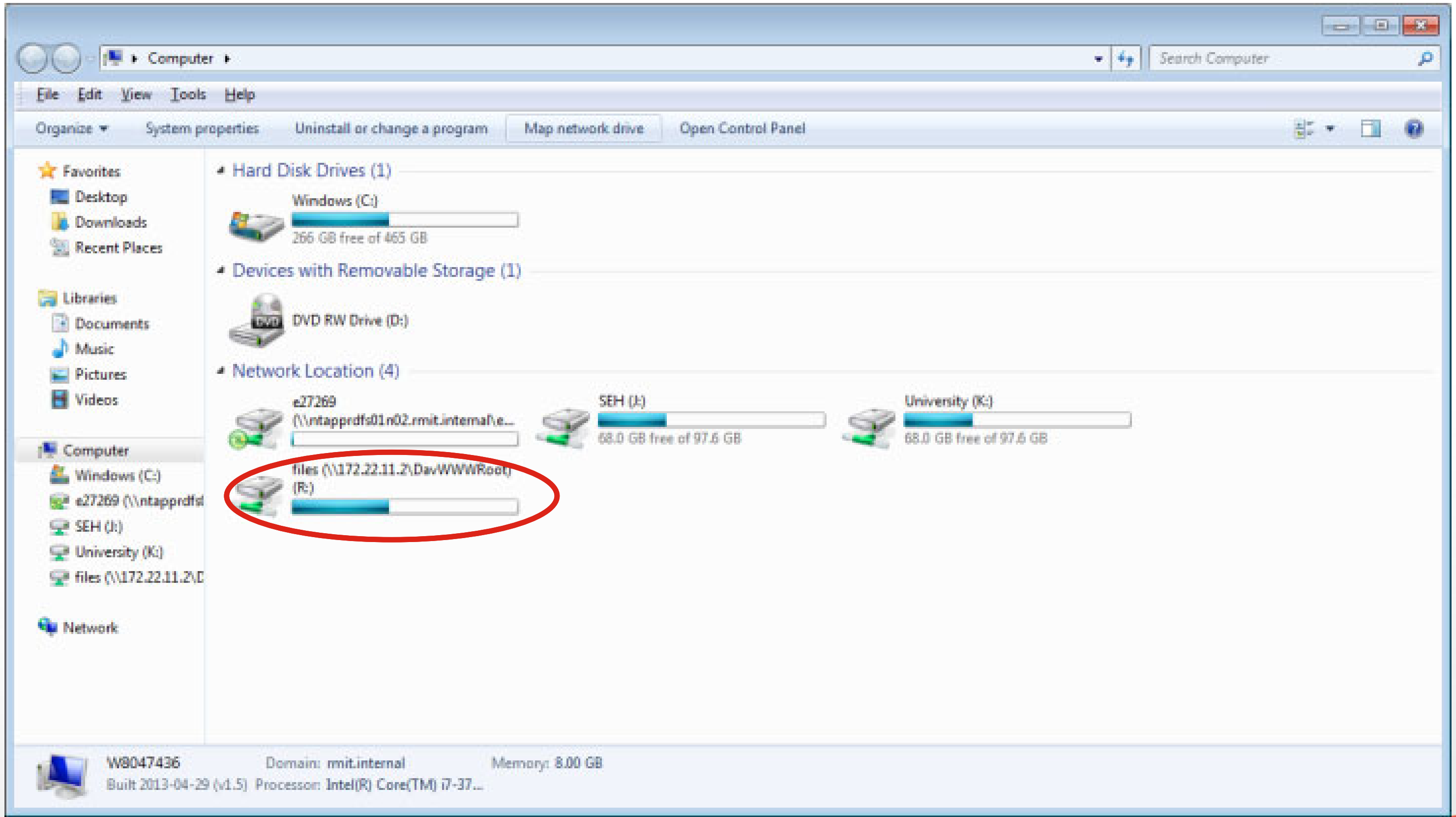
Saving Measurement Data in myRIO

- After getting the file, you can copy it from myRIO into your PC.



Saving Measurement Data in myRIO

- When you go to Computer again, you will see that a new drive (R) is created.

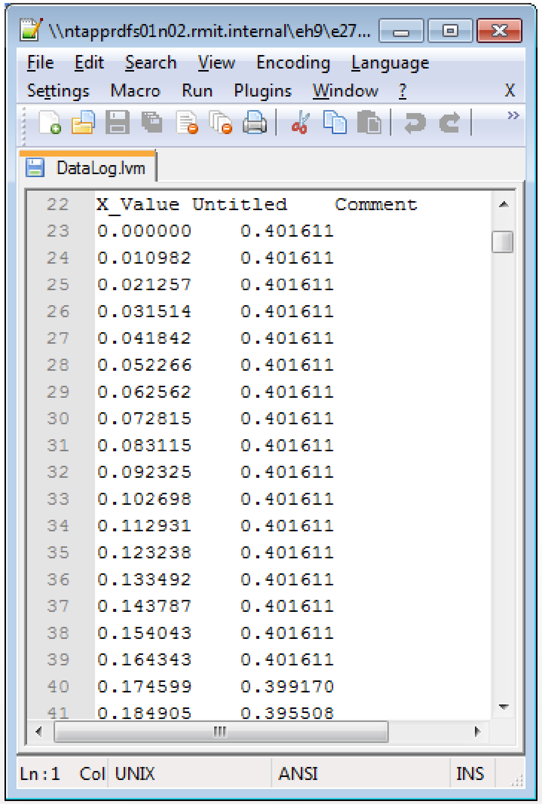


Saving Measurement Data in myRIO

- In case the network drive feels slow, e.g. when opening folder, adding file etc.
  - Go to Control Panel
  - Select Network and Internet
  - Choose Internet Option
  - Under “Connections” Tab, click LAN Settings
  - Uncheck Automatically Detect Settings

Software Timing vs Hardware Timing

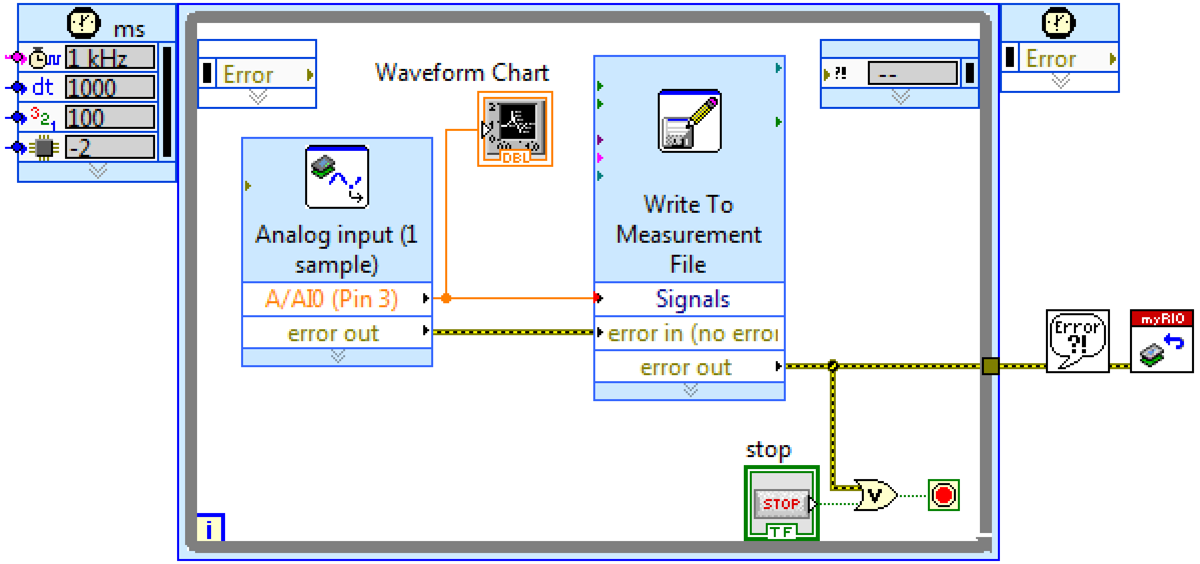
- One thing you would notice is that the x_value (time) is not exactly 10ms apart.



- This is because the while loop with “wait (ms)” is only a software timer and is not absolutely accurate.

Software Timing vs Hardware Timing

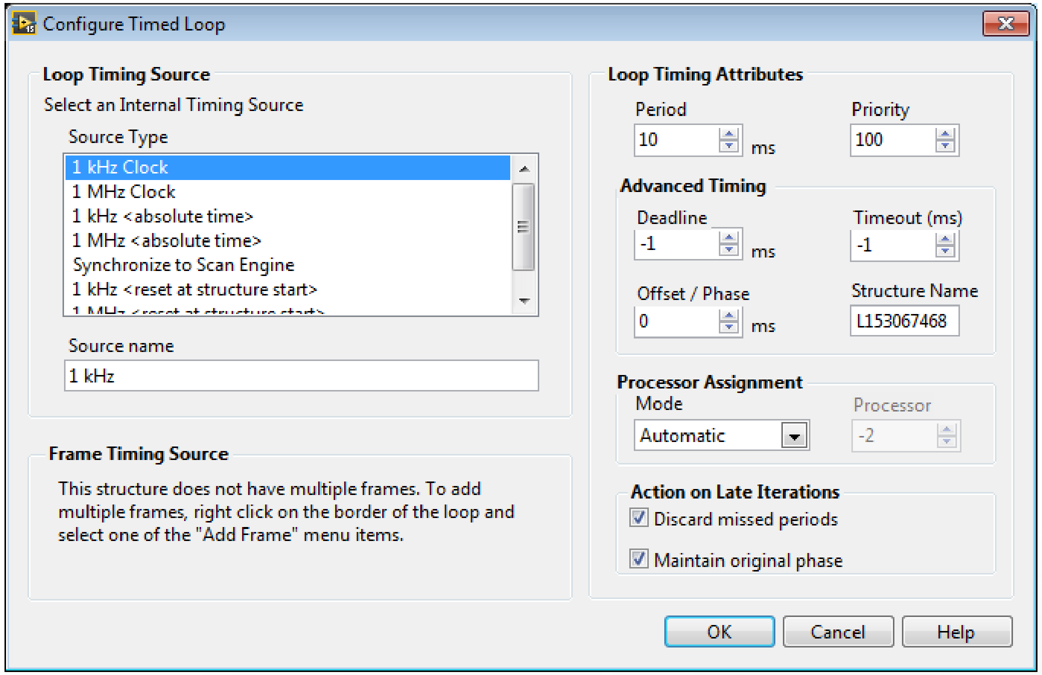
- Now, pop up on the while loop and click “Replace with Timed Loop”.
  - Remove “wait (ms)”.



- The timed loop uses hardware clock and is accurate.
- Also, timed loop has higher priority than other parts of the VI, thus guaranteeing real-time performance.

Software Timing vs Hardware Timing

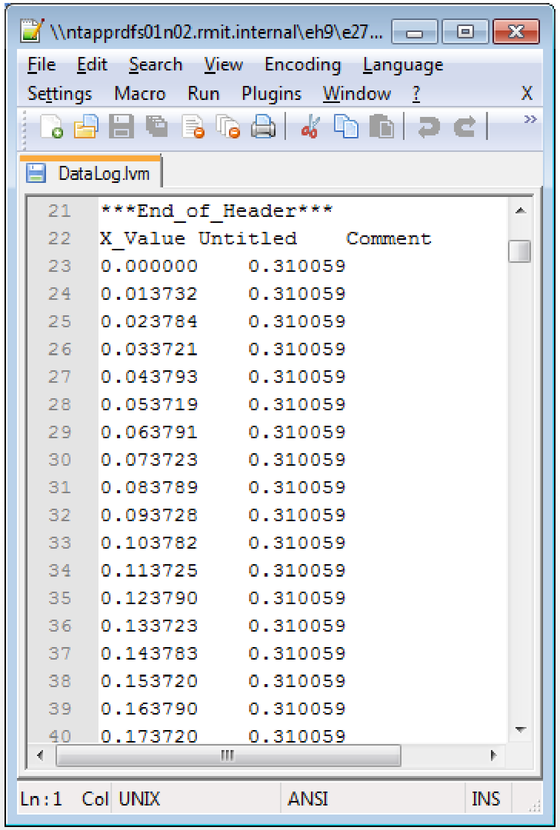
- Double click on the top left box, and you will see the following dialogue.



- We leave the timing source as 1 kHz.
- Change the period to 10ms.
- Priority (here 100) means the priority of THIS timed loop relative to other timed loops, if exist.

Software Timing vs Hardware Timing

- Run the VI again, and get the file from myRIO SS hard disk drive.
- Now, the x-values are much more consistent with the desired 10ms interval.




```
***End_of_Header***
X_Value Untitled    Comment
0.000000    0.310059
0.013732    0.310059
0.023784    0.310059
0.033721    0.310059
0.043793    0.310059
0.053719    0.310059
0.063791    0.310059
0.073723    0.310059
0.083789    0.310059
0.093728    0.310059
0.103782    0.310059
0.113725    0.310059
0.123790    0.310059
0.133723    0.310059
0.143783    0.310059
0.153720    0.310059
0.163790    0.310059
0.173720    0.310059
```

Thank you!

Reminder: Please redesign gripper before next week's class.
Save ALL the individual parts into .STL format

Any Questions?

RMIT UNIVERSITY

Content

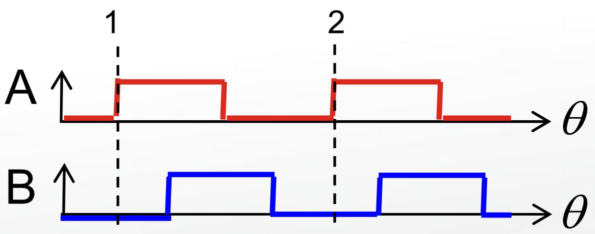
- Position Measurement
    - Measure using myRIO
- Speed Measurement
- Vibration and Acceleration Measurement
- Stress and Strain Measurement
    - Measure using myRIO
- Saving (Sensor) Data using myRIO
- Attachment: Writing FPGA Codes

Attachment: Encoder - FPGA

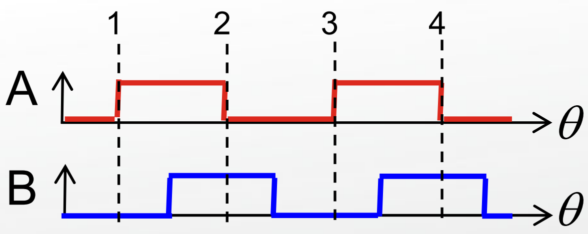
- After getting to know the basics of coding in FPGA, let's do something more exciting.
- We will create a code to read in encoder signals (in digital form) to:
  - Calculate the distance travelled
  - Determine the direction

Recall: How to Use Pulse Signals

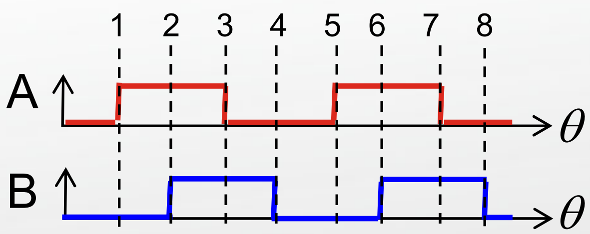
- How to determine distance travelled? – By counting the edges!
  - E.g. from datasheet, the encoder gives 100 PPR (Pulses per revolution).
    - If use only rising edge of channel A: 100 PPR → 3.6° per pulse.
    - If make use of both rising and falling edges on channel A, the resolution is increased by two (x2 logic): 3.6° / 2 = 1.8° per pulse.
    - If makes use of both rising and falling edges on both channels, the resolution is increased by four (x4 logic): 3.6° / 4 = 0.9° per pulse.
    - Cumulative sum of the counts = total distance travelled (Position).



x1 logic



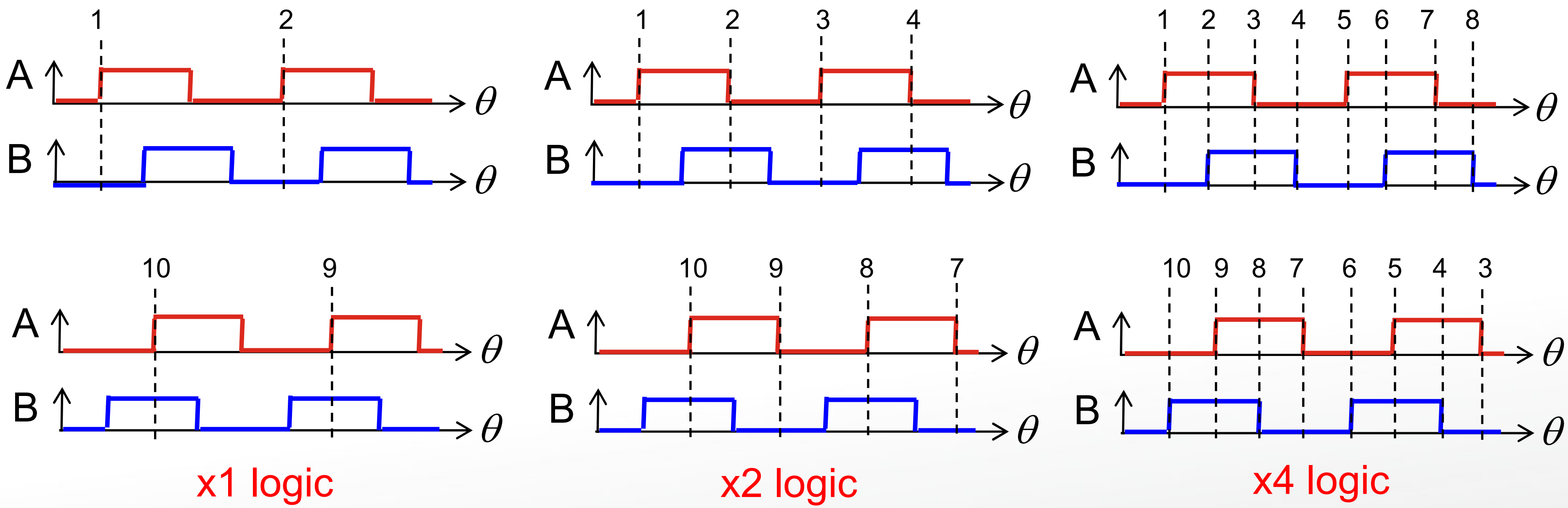
x2 logic



x4 logic

Recall: How to Use Pulse Signals

- How to determine direction? We know that:
  - for CW rotation, B is delayed.
  - For CCW rotation, A is delayed.



- Detect rising edge of one channel, and check status of the other channel.
  - E.g. in x4 logic:
  - A (now) not equals B (just now) → CW
  - A (now) equals B (just now) → CCW

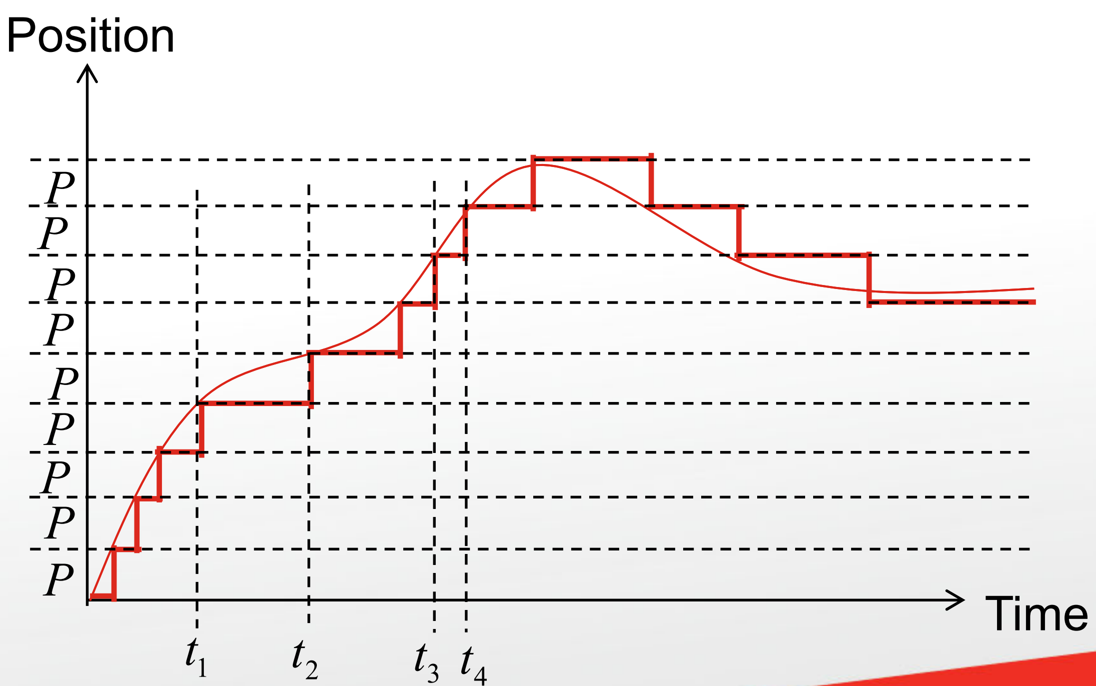
RMIT UNIVERSITY

Recall: Deriving Speed from Position

- There are two ways to calculate speed from position:

  a) Make use of the constant quantization level P:

  - Determine the time duration between position increment.
  - Then $\text{Speed}_1 = \frac{\Delta p}{\Delta t} = \frac{P}{t_2 - t_1}$ $\quad \text{Speed}_2 = \frac{\Delta p}{\Delta t} = \frac{P}{t_4 - t_3}$ → constant



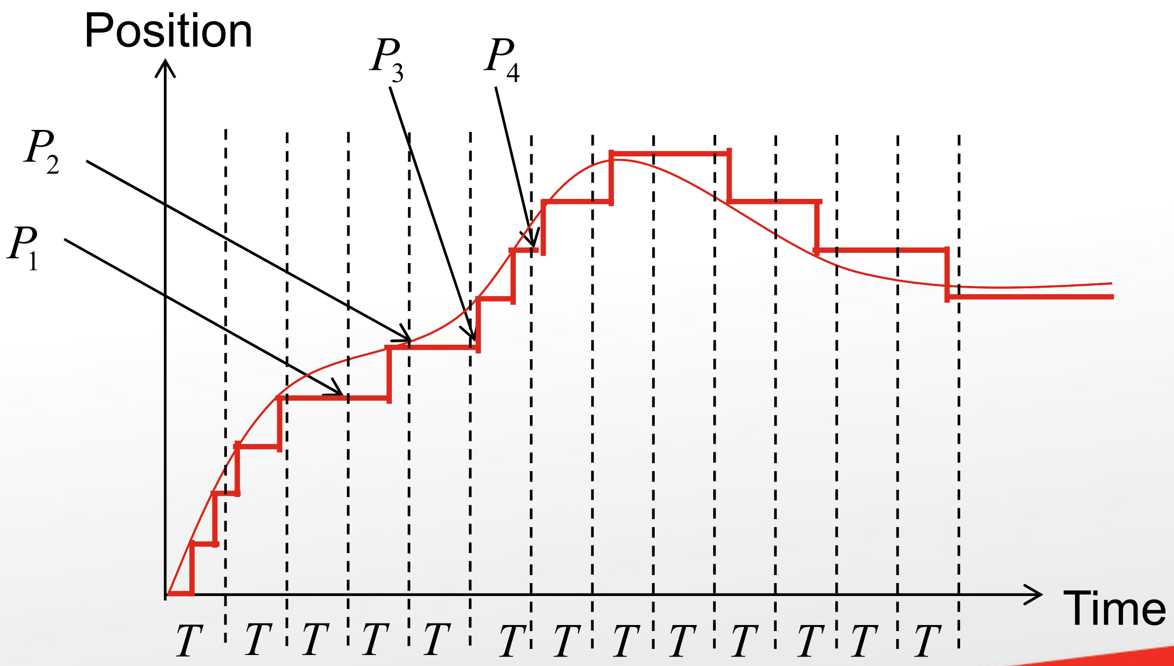
a)
Good for
low speed

Recall: Deriving Speed from Position

- Second method:

  b) Make use of the constant sampling interval T of the discrete-time system.

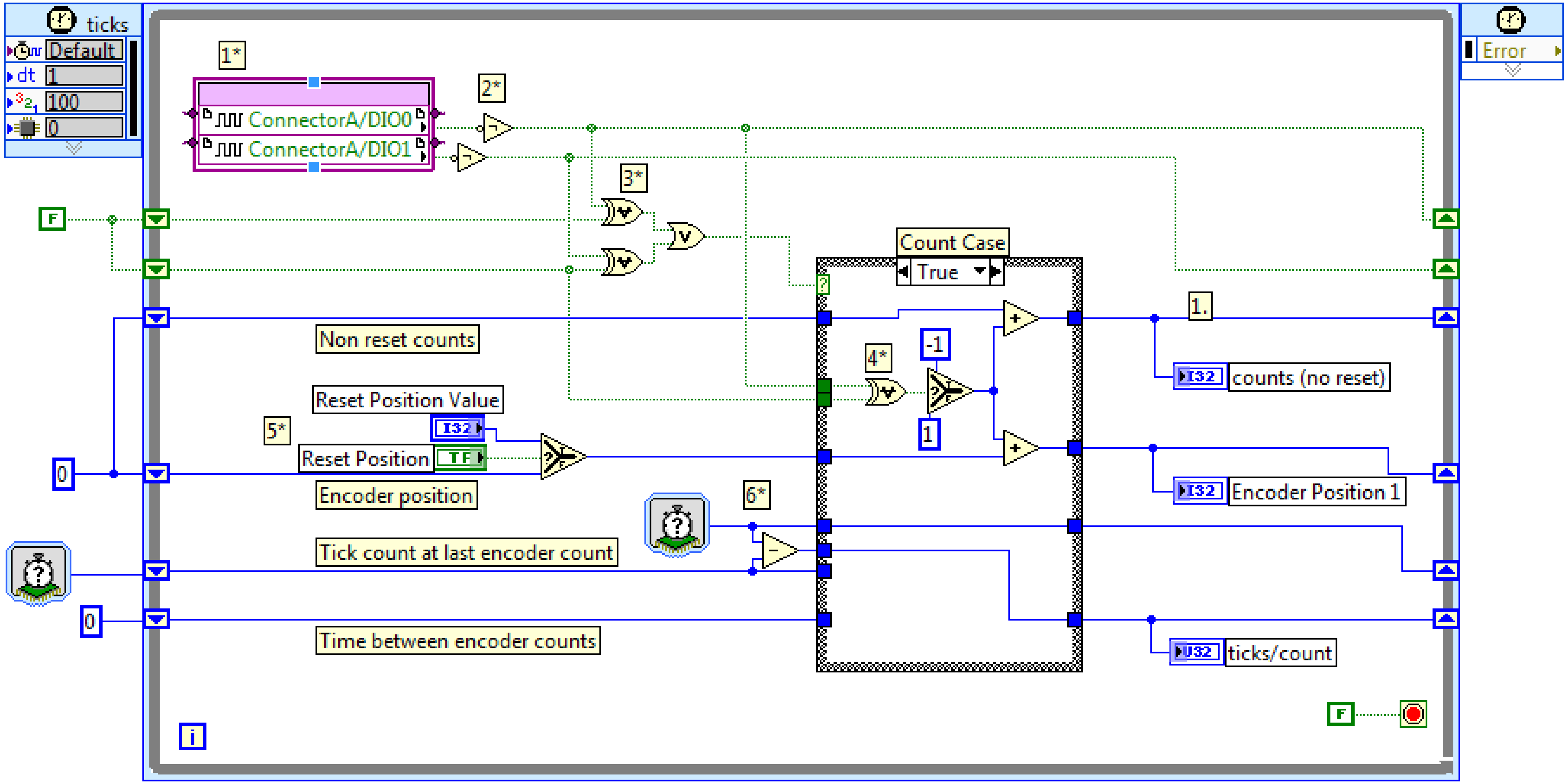
  - At the sampling time instance, get the P value.
  - Then $\text{Speed}_3 = \frac{\Delta p}{\Delta t} = \frac{P_2 - P_1}{T}$ $\text{Speed}_4 = \frac{\Delta p}{\Delta t} = \frac{P_4 - P_3}{T}$ → constant



b)
Good for high speed

Encoder - FPGA

- As detailed in last week's attachment, please prepare your project to include RT.vi and FPGA.vi.
  - Project name "EncoderFPGA".
- In FPGA.vi, create the VI as shown in the next few slides, and compile when done.



1)
Read the current encoder phase states:
DIO0 - Phase A
DIO1 - Phase B
DIO2 - Index

2)
Invert due to Pull-up resistor

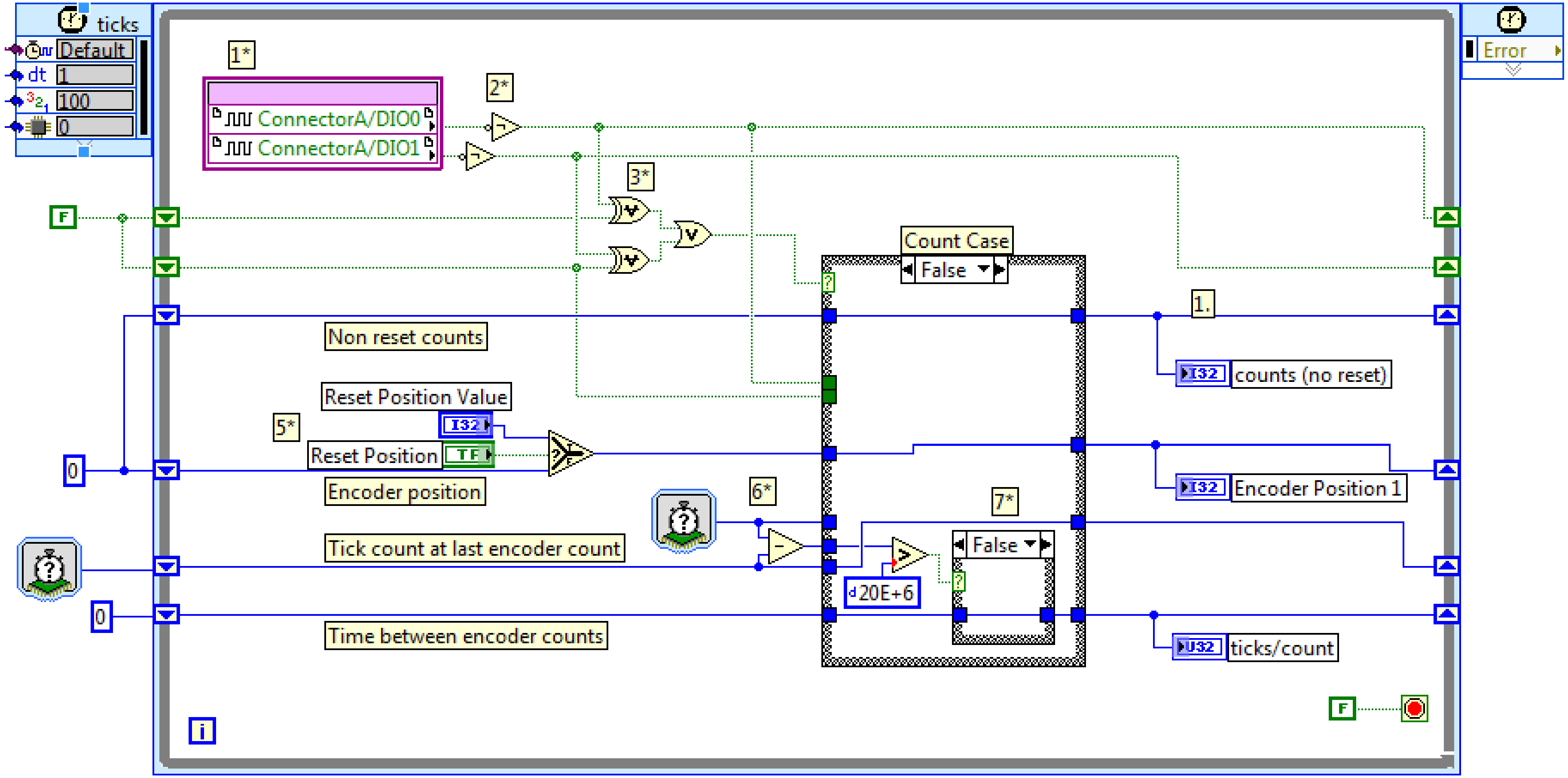
3) XOR: detect edges.
If A is not the same as B at this step, it means there is an edge

4)
XOR: A (now) vs B (just now).
If same, then CCW
If different, then CW

5)
When Reset Position is pressed, Encoder position resets to the "Reset Position Value"

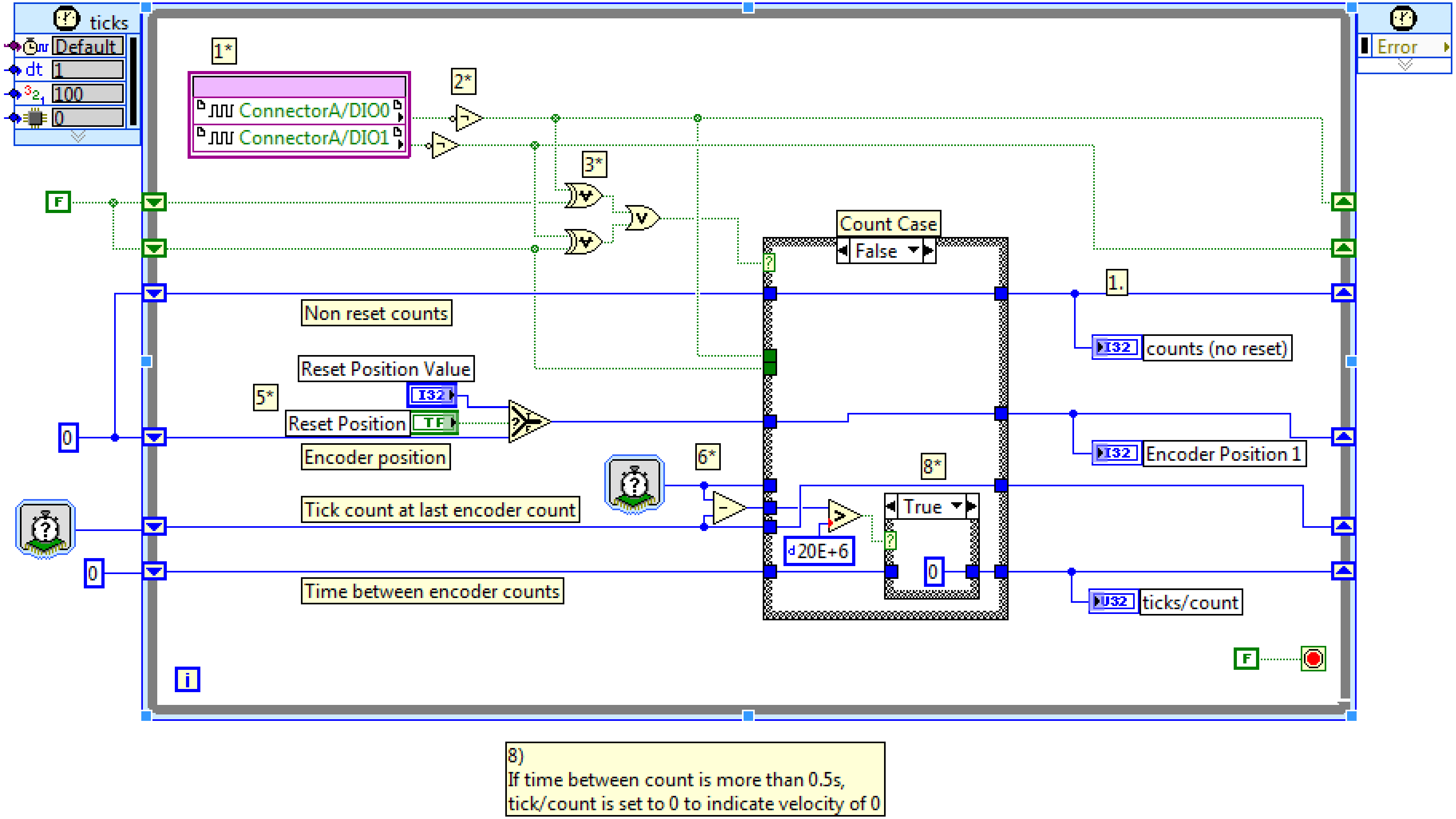
6)
When edge is detected, calculate the difference between current tick count and previous tick count, to calculate speed

Case 1: Edge detected



7)
If time between counts is less than 0.5s (20E+6 ticks * 25ns),
continue passing through the same ticks/count which was calculated at the instant of edge detection

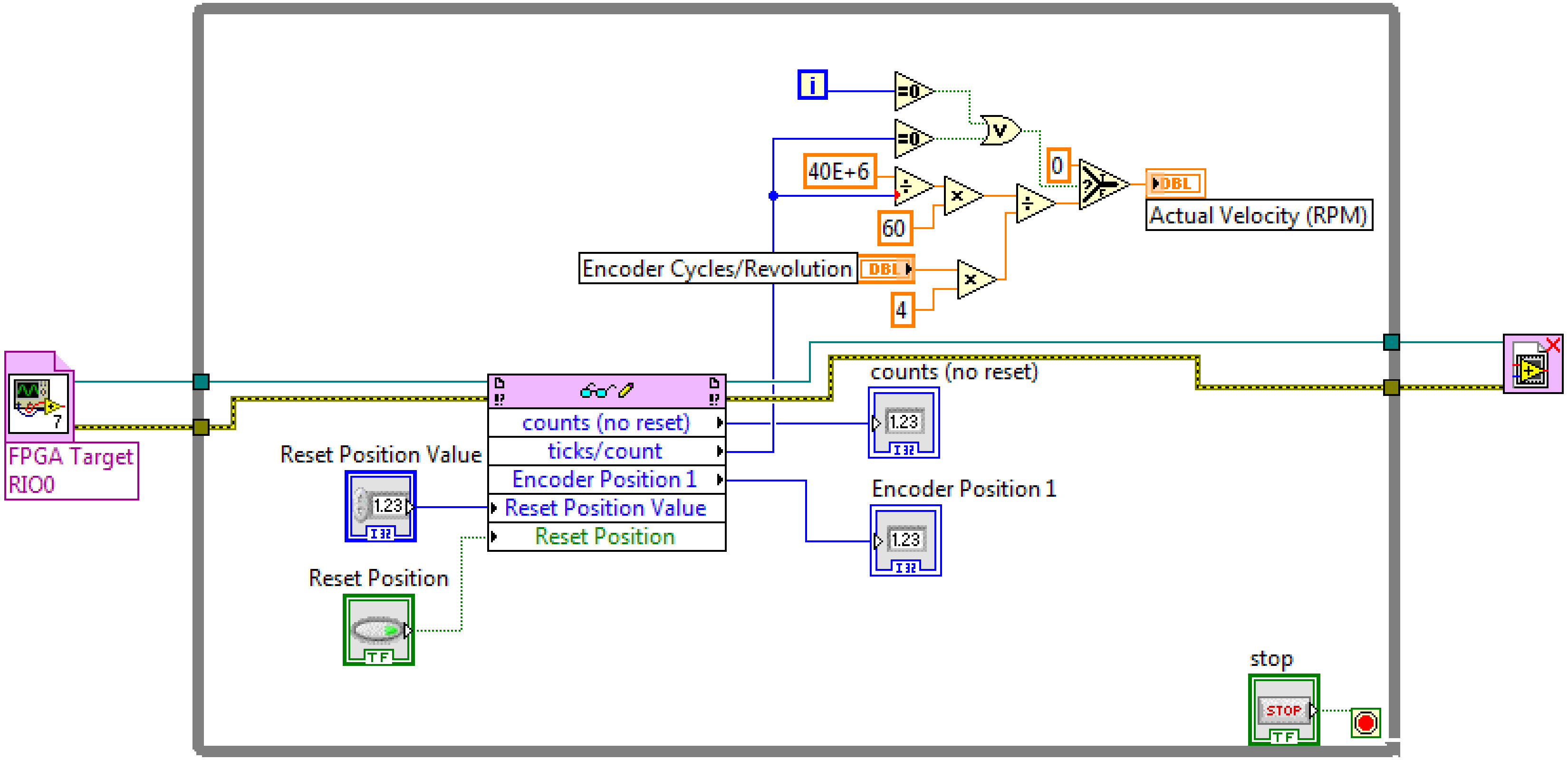
Case 2: Edge not detected, but previous time between edge detection is less than 0.5s



Case 3: Edge not detected, but previous time between edge detection is more than 0.5s

Encoder - RT

- Create the VI as shown:



- Run the VI and you will be able to see the encoder signal.

Thank you,
Questions