ElCoM
ELECTRICAL COMPUTER MECHATRONICS

تقدم لجنة ElCoM الاكاديمية

تلخيص لمختبر:

أنظمة ميكاترونكس

جزيل الشكر للطالبة:

سارة آبو سارة

Exp(1): Introduction to LabView

LabView ≡ Laboratory Virtual Instrument Engineering Workbench

LabView programs are called virtual instruments, or VI's, because their appearence and operation imitate physical instruments, such as oscilloscopes and multimeters.

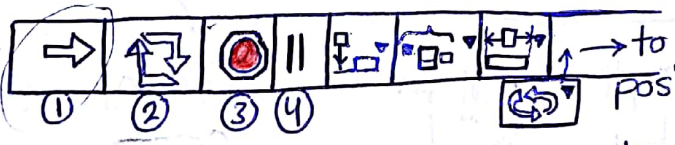
It contains a comprehensive set of tools for → acquiring
analyzing
displaying
and storing data

as well as, tools to help you troubleshoot code you write.

VI Components →

1] Front Panel: - it's the user interface of the VI.
الشاشة الأمامية
- we build it using an interactive input and output terminals of the VI, called [Controls and indicators].
يطلع من البرنامج — يعرض على البرنامج

- Controls, simulate instrument input mechanisms and supply data to the block diagram of the VI, such as [knobs/pushbuttons/dials/...]
- Indicators, simulate instrument output mechanisms and display data the block diagram acquires or generates, such as [graphs/LED's/...]

Front Panel Toolbar ⇒ 

→ to control the size & position of objects on the front panel

① Run Button to run a VI, appears in many forms according to the VI :

- solid white arrow indicates you can use the VI as a sub VI if you create a connector pane for the VI
- يظهر هذا الشكل while the VI runs If the VI is a top level VI, meaning it has no callers, so it isn't a sub VI
- appears when the VI that is running is a sub VI
- ↳ appears broken like that when the VI you are creating or editing contains errors. (block diagram wiring) يظهر هذا الشكل عندما يكون هناك خطأ في البرنامج the VI is broken and can't run.

Error List لمعرفة الأخطاء

② Run Continuosly Button → to run the VI until you abort or pause execution.

you also can click the button again to disable continuos running.

③ Abort Execution Button → to stop the VI immediately if there's no other way to stop the VI.

If more than one running top level VI uses the VI, it will be dimmed.

④ Pause Button → to pause a running VI

when you click it again, LabView highlights on the block diagram the location where you paused execution and then the VI will run again.

التدريس خلفة موبين

2] Block Diagram: Contains the graphical source code (G-code)

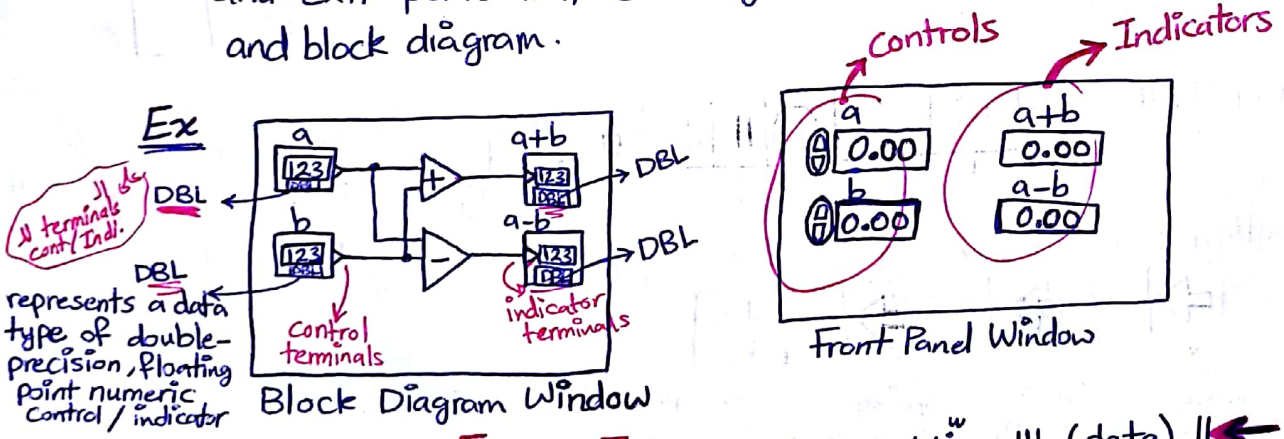
الشاشة اللي بنبني فيها الكود

represents by functions to control the front panel objects. [front panel objects appear as terminals on the block diagram].

أي شي بحطه بالـ front panel بيظهر ما بيناظره على الـ block diagram لكن العكس غير صحيح ..

Block diagram objects ⇒

① Terminals → represent the data type of the control or indicator, they are entry and exit ports that exchange information between the front panel and block diagram.



"Block Diagram" بتتصل إلى الـ [front panel] بالـ controls اللي بدخلها على الـ (data) ← بواسطة الـ (Control Terminals) ..

هاي الداتا بتشتغل على الـ (+/- functions) وبطلع منها بالآخر نتائج حسابات produce new values flow to the indicator terminals, where they update the front panel indicators.

② Nodes → objects that have inputs and/or outputs and perform operations when a VI runs.

cluding (sub VI، Statements، operators، functions، constants، express VI، structures) + shift Reg.

↳ they're graphical representations of the loops and case statements, used on the block diagram to repeat blocks of code and to execute it conditionally or in a specific order.

③ Wires → to connect the control and indicator terminals to the other functions. each one has a single data source, but we can wire it to many VI's and functions that read the data.

They're comes in different colors, styles and thicknesses, depending on their data types.

Wire Type	Color
Numeric	Orange [floating pt.] → يعني فيها كسور عشرية / Blue [integer] → الاعداد الصحيحة
Boolean	Green
String	Pink

Floating pt.: double and singles that are a signed numbers with a decimal component

Integer: signed or unsigned

Boolean: have 2 values (T/F)

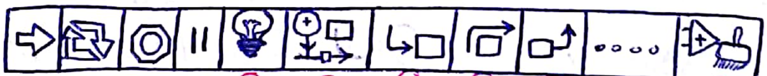
String: sequences of characters

Clusters → groups of various data types and indicated by a thick brown data wire

- A broken wire appears as a dashed black line with a red X in the middle occurs for a variety of reasons, such as when you try to wire 2 objects with incompatible data types.

* Error Cluster
Composed of: • boolean status
• numeric error code
• string source
used to indicate warnings and errors.

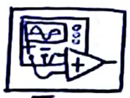
Block Diagram Toolbar ⇒



① highlight Execution
② Retain wire values
③ Step Into
④ Step over
⑤ Step out
⑥ Clean up diagram

3 Icon and Connector Pane →

after we build a VI front panel and block diagram, building the icon & the connector pane use the VI as a sub VI. It corresponds to the function prototype in text-based programming languages.

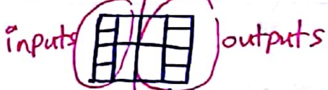


Icon

شغلي يميز الـ VI logo

[Graphical Representation for the VI]

it doesn't affect the VI functionality



Connector Pane

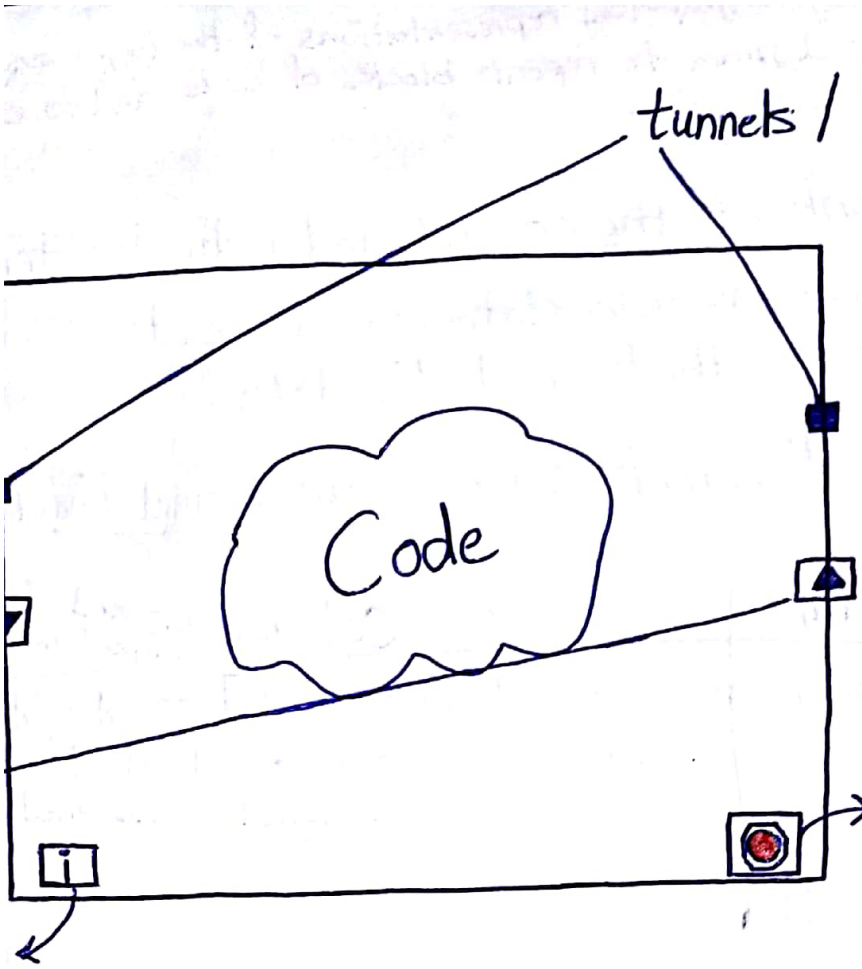
it's a set of terminals that correspond to the controls and indicators of that VI. وظيفتين ↓

① defines inputs and outputs that you can wire to the VI, so you can use it as a sub VI.

② receives data at its input terminals and passes the data to the block diagram code through the front panel controls

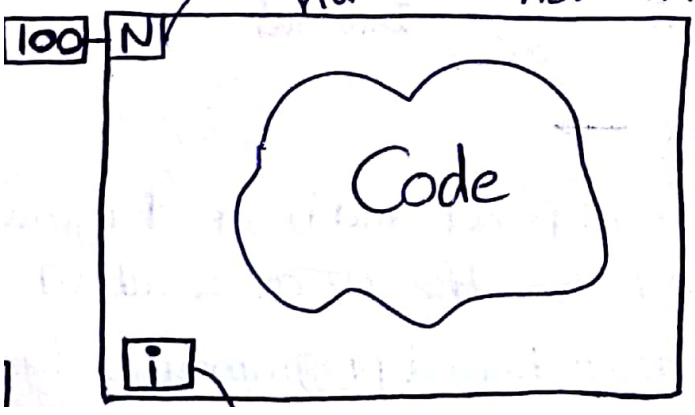
③ and ... receives the results at its output terminals from ... panel indicators.

Scanned by CamScanner



* while loop executes the code until a condition occurs
* it always executes at least once.

- أي شيء في اللوب لازم أخذ قراءة واحدة فقط



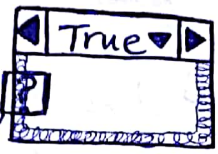
الفكرة

انه اللوب هاي بتخليها تنفذ الكود لحد ما تصير عدد الـ iterations مساوية لـ N

times unless a conditional terminal is added | only if the value at the conditional terminal meets the condition

- Can execute zero times | - Must execute at least once
- Tunnels automatically output an array of data | - Tunnels automatically output the last value.

③ Case Structures :



It has 2 or more cases. Only one is visible at a time, and the structure executes only one case at a time.

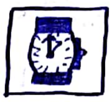
Selector terminal
input value/selector متشابك معها to determine which case to execute

- if the selector terminal data type is boolean → the structure has a true & false cases
- " " " " " " " string integer or enumerated type value → " " can have any no. of cases.

Timing VI's :

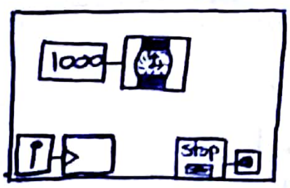
① Wait functions → uses the ms (milliseconds) clock of the operating system

a.) Wait (ms) :
waits until the ms counter counts to an amount equal to the input you specify.



* when Labview calls a VI for example:
If millisecond timer value = 112 ms (وقت الVI)
10 ms to wait ← ودخلت ع التأخير
the VI will finish when ms timer value = 122

*



مع البلاي صيصير البرنامج يتنفذ
ثانية وحدة في كل مرة
وبين كل لوب والثانية في ثانية بتاعتهم
بالتالي وقت ال VI كامل حيزيد عليه بعدار 1000 ms كل مرة ← 1500، 2500، ...

waits until the value of the ms timer becomes a multiple of the specified ms multiple (المدخل ع التايمر)

* for example → if ms multiple (الوقت اللي بيدخل ع التايمر) is 10ms, and ms timer value = 112 ms the VI will wait 8 more ms until the ms timer value = 120 ms a multiple of 10

أكبر من 112 ms لهيك 10 ms

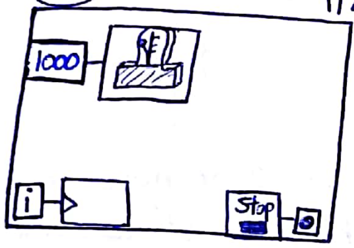
20 30 40 50 120

120 − 112 = 8 ms

حتى يوصل البرنامج لل (120 ms)

التايمر بعطيه 8 ms يحل عليه ازا كان بالأول مقداره 112 ms

*



لما نيجي نشغل البرنامج فلاقيه بنفذ اللوب بأقل من الثانية

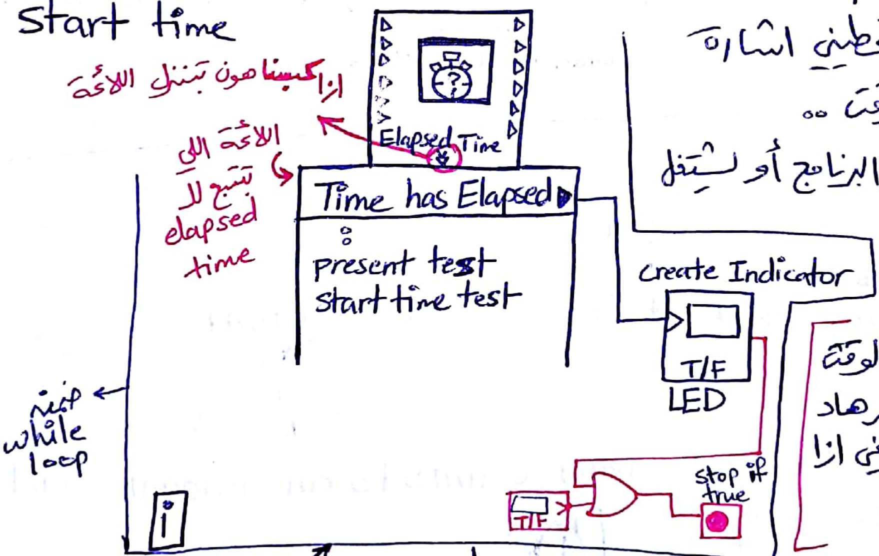
فالتايمر هون دوره يعطي ديلاي بمقدار معين ع الساعة الداخلية للـ operating sys. لكل loop على 1000 ms أو مضاعفاتها

② Elapsed Time →

indicates the amount of time that has elapsed since the specified start time

بدخل عليها وقت تعطيني اشارة ازا خلص هاد الوقت ..

بعدها الوقت بطفي البرنامج أو بشتغل alarm معين



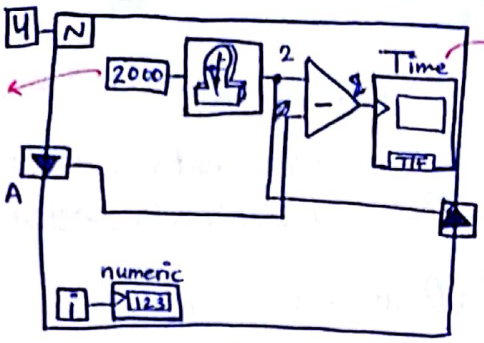
بيضوي الليد ازا الوقت اللي دخلته عالتايمر هاد خلص وبضل طافي ازا لسه ما خلص.

* لو مثلاً بدي أعمل بعد 3 sec. يضوي الليد ويطفي اللوب ←

بعدها أدخل عليه ال 3 sec لل تايمر

∴ هاد التايمر بعطي ديلاي بمقدار Δ ، وهاي ال Δ بتساوي الفرق بين الرقم اللي دخلته ع التايمر والزمن تبع ال operating system

لكن ازا كان الزمن تبع ال operating sys. أكبر منه الرقم اللي دخلته ع التايمر فيصير التايمر يحل عليه بناءًا ع مضاعفاته



0.9s

25

29/4

2-0.9 = 1.1

2-0 = 2

-2 = □

① What's the name of the structure? For loop

② What's the name of "A"? Shift Reg.

③ Without the presence of the time function, the single iteration takes 0.9s to execute, what's the value of "time" after the 2nd iteration?

in the 1st iteration: ② - ∅ = 2 → 2 with it presence

" " 2nd " : ④ - 2 = 2 → 2.9 with it presence

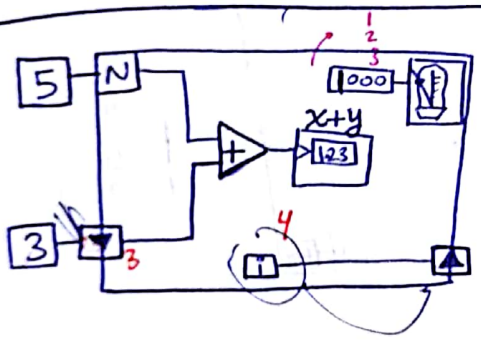
④ How many seconds does the program take?

4 * 2 = 8 sec.

⑤ What's the value of "numeric" at the end of the program? 3

↳ i = N-1

Q5



① how many iterations will the program execute? 5

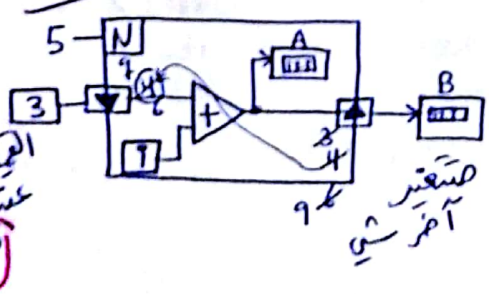
② what's the value "3" called? initial value for the shift reg.

③ If the single iteration requires 1.1 s without the presence of the time function. How long will a single iteration take with its presence? [2 sec.]

④ how much delay does the time function provide? delay = 2 - 1.1 = 0.9 sec.

⑤ What is the value of "x+y" at the end of the program? 5+3 = 8

Q6



i	A	B
0	3	--
1	3+1 = 4	--
2	4+2 = 6	--
3	6+3 = 9	--
4	9+4 = 13	13
5	--	--

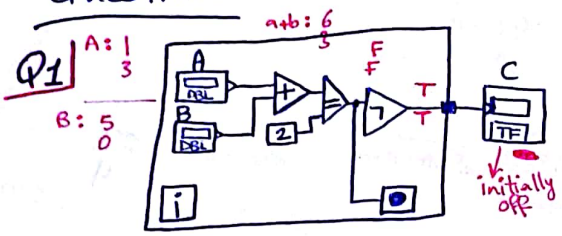
If N=5 initial = 1

N	C
0	--
1	--
2	--
3	--
4	32

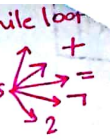
A: 2, 4, 8, 16, 32

Questions

Q1



① How many nodes ? 5 nodes



② C represents indicator

③ i represents Loop Iteration no.

④ assume that "C" is initially off, and the program is run once, fill :

A	B	C
--	--	off
1	5	off
3	0	off
2	0	off
2	2	--

1+5 ≠ 2 ←

3+0 ≠ 2 ←

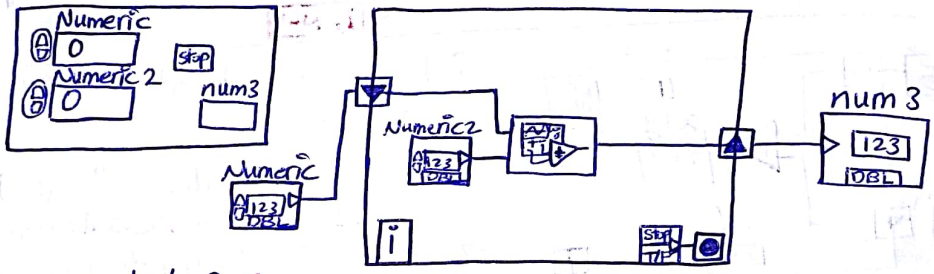
موقف من اللوب وما تنفذ هاي ←

initially on كيف راح يكون

A	B	C
1	5	on
3	0	on
2	0	off
2	2	--

أي شي برا الـ while راح يحل update آخر البرنامج

Q2 Based on the front Panel and block diagram below, answer the following ?



① how many controls ? 3

② how many indicators ? 1

③ how many nodes ? 3

Q3

① Is this a front panel or a block diagram ? block diagram

② What's the name of the structure ? while loop structure

③ which one is a control "C" or "A" ?

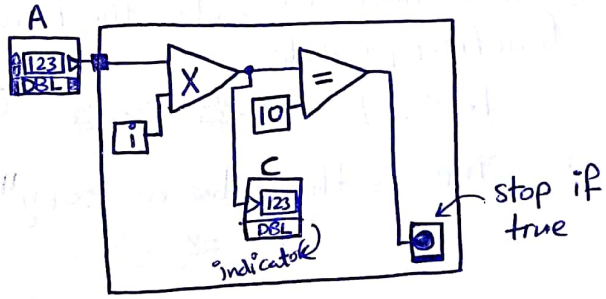
④ If A is set to 3, after how many iterations will the program end ?

∞

⑤ if A is changed to 1 after the 2nd iteration, what's the value of "C" [during that iteration]

C = 3

3 صفر عشرها لأنها بتزل اللوب



3*0 = 0 → C=0 F

3*1 = 3 → C=3 F

3*2 = 6 → C=6 F

3*3 = 9 → C=9 F

3*4 = 12 → C=12 F

Exp(2): Data Acquisition [DAQ]

عبارة عن (interface device) فيه pins مخصصة لـ digital/analog/GND/Vcc
من الـ environment اللي اماميّ [Counter signals] بتقدر تجيب المعلومات وتقرأها..

Sensor —(بقرأ منه)→ Voltage
→ current

Signal → Input / → output → analog (current/voltage) / → digital / → Counter

[Sensor] —(أخذ منه قراءة)→ [Interface DAQ hardware] → [Software programming]

Part 1) Analog Input ⇒

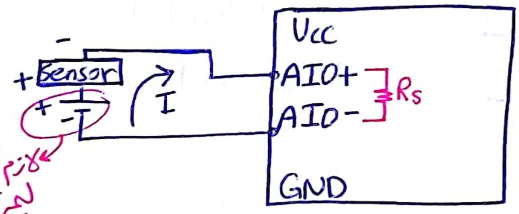
[1] Connecting analog current inputs →

[الـ Sensor بيعطي الـ DAQ Analog Current]

* Differential Measurements (Current Loop Measurements)

Most devices can measure ±20 mA

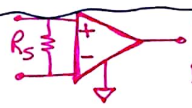
مشان أعمل current loop



لازم أعطيه power supply ليمر تيار

AIO+ / AIO- → channel عبارة عن واحدة لـ 2 pins

R_s ≡ Shunt Resistor

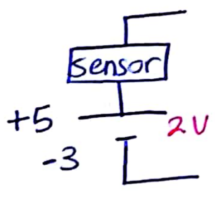


PGIA: Programmable Gain Instrumentation Amplifier

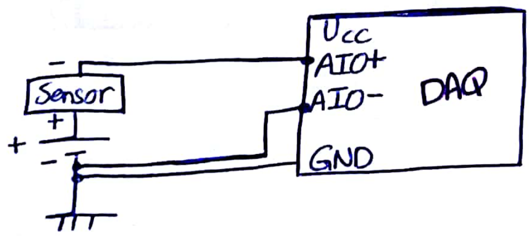
→ ① Used to amplify low voltages on analog inputs

→ ② to provide variable gain which is controlled in software.

بس هون ما في common GND
لأنه البطارية ممكن يكون طرفها ≠ صفر السالب

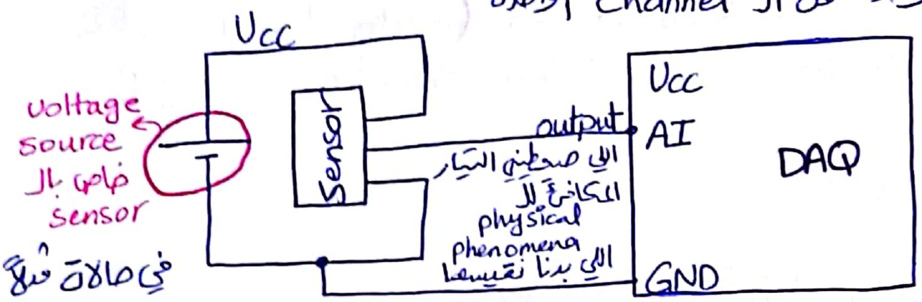


فلهيك لازم نعمل (common GND)



* Single Ended Measurements (3 wire_current) → Transducer

يعني بستخدم pin وحدة من ال channel الواحدة



في حالة انو اما ال sensor لبشتغل ع 10V وال Internal DAQ Vcc بتعطي بس لغاية 5V فبهالحالة رح أضطر ل external power supply خاص لاحتياجات ال sensor التشغيلية وحسب التيار اللي رح يسحبه ال sensor لياخذ قراءة

---

2 Connecting Analog Voltage inputs →

Source (الفولتية اللي أعطاني اياها ال sensor) / source لل DAQ — Measurement (DAQ) نوع ال DAQ اللي بدي أقرأ منه

- floating [طرفه السالب ≠ صفر]
- Grounded [طرفه السالب = صفر]

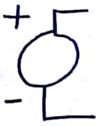
- Single Ended
- Differential

الحصلة ← 4 circuits combination

اما ال sensor بتكون floating وبشبكه ع صورة Single Ended أو ع صورة Differential

أو ال sensor بتكون Grounded وبشبكه Single Ended أو Differential

Floating output voltage Sensor →

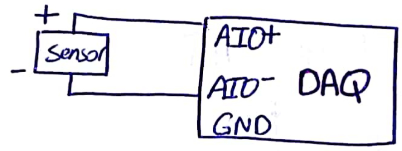


+ ← 2pins طالع منه
- ←

← ولا واحد منهم قيمته صفر
بيعطيني الفرق بين قيمتين
مابعرف أي طرف فيهم موجب وأيهما
سالب ، لكن بعرف مين الأكبر من
الثاني (no zero volt)
no GND

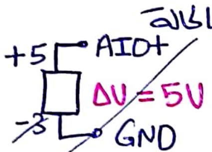
Sensor بحدد كيف بدي أوصل هاي ال
وحسب نوع الـ DAQ
Single Ended
Differential

a.) floating sensor + diff. DAQ :

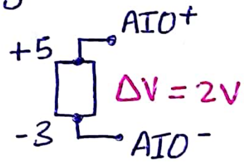


← ما بصير أشبك الطرف السالب مع الـ GND لأنه مربوط بـ Floating بالحالة.

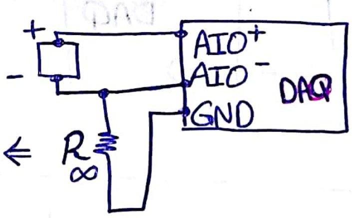
Channel ∅
إلها 2 pins بالـ DAQ
اللي + اشتريتها
اذن مكون نوعها
(differential DAQ)



رح تعطيك قراءة مختلفة

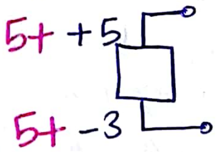


← لكن لابد من طريقة لتوصيل GND :

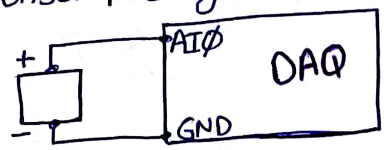


← مقاومة كبيرة جدًا للتيار (بتصير كأنها open CCT)
وكل الـ drop voltage بيكون عليها

وبالتالي قدرة أخفف الـ noise فيها ولو ما حطيت R∞
(بغض النظر عن الخطأ اللي رح يصير في الـ data) برضو ما بيكون في noise
لأنه differential عم بقيس فيها الفرق بين قيمتين
لو زدت عالطرفين (5V) مثلاً مبتكنسلوا مع بعض ..



b.) Floating Sensor + Single Ended DAQ :



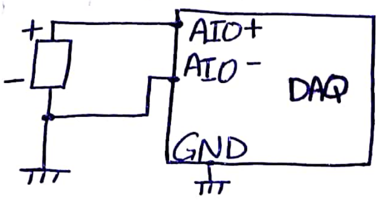
[مفضلة أكثر للاستخدام]

الـ (noise) فيها عالي ولكن حجمها صغير لأنه الـ channel الواحدة مكونة من (one pin) AI
بعكس الـ differential DAQ اللي بيكون حجمها الضعف لأنه
الـ channel الواحدة فيها عبارة عن 2pins → AI+ , AI−

Grounded Sensor → هو اللي بيكون الطرف السالب منه (ØV)

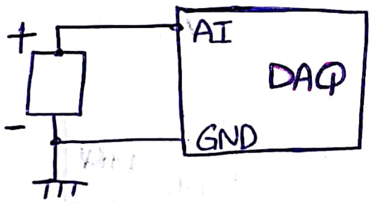
a.) Grounded Sensor + Diff. DAQ :

الفرق ← ما نحط مقاومة كبيرة قبل الـ diff. DAQ + floating voltage



[مفضلة أكثر للاستخدام]

b.) Grounded Sensor + Single Ended DAQ :



(أسوء نوع)

Because, it's a power wasteful due to the ground loop.

Part 2) Analog Output ⇒

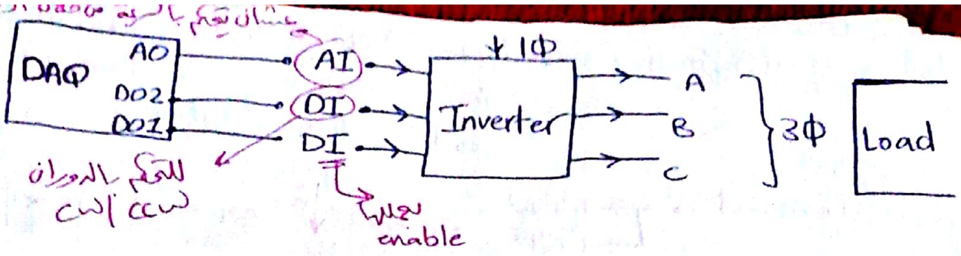
الـ Card بالحالة بده يعطي سيجنال للـ Load

بركّب اللود مباشرة مع الـ DAQ بالطريقة إذا كان compatible مع الـ DAQ و الـ DAQ قادر يشغل اللود

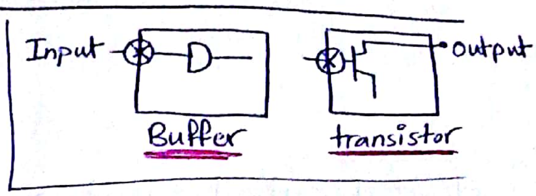
لكن ببعض الحالات محتاج الـ interface مثلاً لما بدي أركّب 3Φ motor مع الـ DAQ

1A 5V → بركبه مباشرة بدون interface → 3A, motor

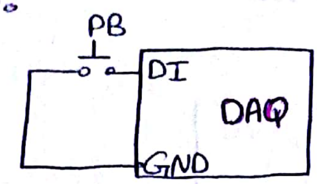
[Interface : like the inverter for example
بحول من DC→AC
1Φ→3Φ]



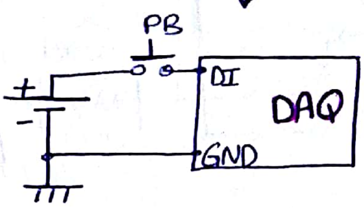
Part 3 Digital ⇒



[1] digital inputs : (DI)



لكن هاي التوصيلة غلط وصفنا فيها الـ Vcc ← external from the DAQ ← منكون غلطة أعلى احالة لأنه الـ digital ما تسحب تيار عالي ..

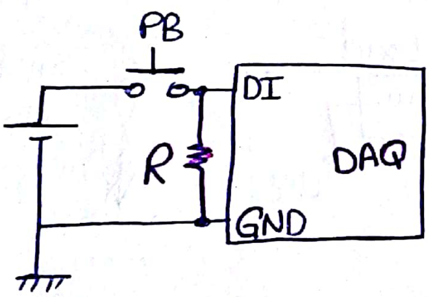


وبدو يعتبر حالة لأنه ما يكون ضاغطة على الـ PB ، ما بعرف كم قيمة الفولتية (unknown) وليس Zero

ولما نضغط على PB حيعطيني فولتية الـ Vcc

لحلها رح أحتاج لـ pull up and pull down resistor

*



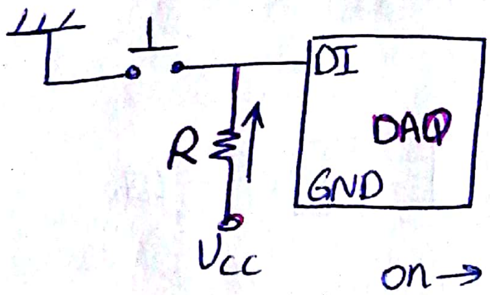
R : pull down resistor

on → Vcc

off → ØV / وليست unknown

[لأنه كل التيار حيروح للمقاومة هاي]

*



R : pull up resistor

لأنها رفعت الفولتية من الصفر للـ Vcc

on → ØV

off → Vcc

(no floating)

بيكون high

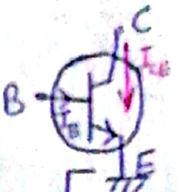
Scanned by CamScanner

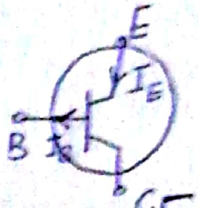
[2] Digital Output : I/P موقسمين digital ال pins تاعون ال

لما بدل ال output بدخل enable ← ال transistors ليمرر تيار

disable ← ال buffer

Transistors → NPN
→ PNP





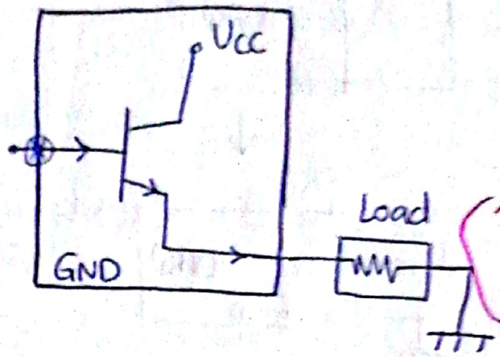
[التيار جاي من اللود ورايح ال transistor]

[اللود بحالة يكون موصول ع الجراوند
التيار طالع من الترانزستور ورايح ع اللود]

digital output types →

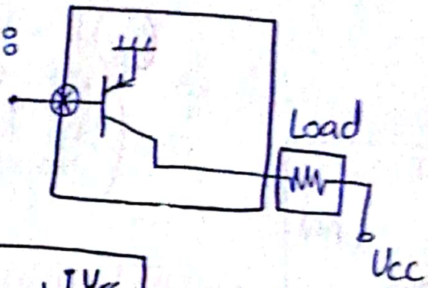
a. Line Driver :

(* اللي مشغّل ال Base صور ال code تبعنا) ⇒



Line Driver : sourcing
اللود مشبوك ع الجراوند

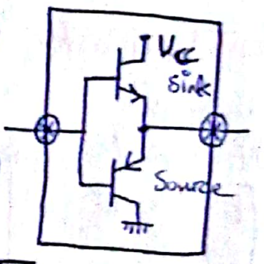
b. Open Collector :



Open Collector : sinking
التيار جاي من اللود ورايح ع الترانزستور

c. Push Pull :

2 transistors
(sink + source)
مستقل حسب الحاجة
الأفضل

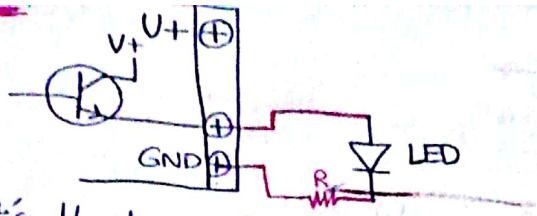


[تم حل مشكلة ال floating]

⇒ Counter I/O
مثل ال digital I/O

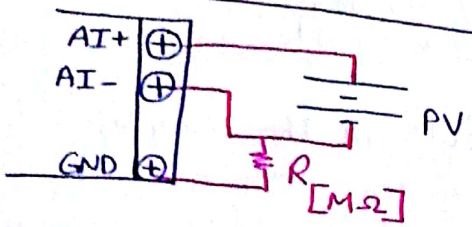
⇒ Signal Conditioning types :

1. amplifiers
2. filtering
3. attenuation
4. Isolation
5. Excitation
6. Linearization



1. What's the type of the digital output configuration? Sourcing
2. Connect the load to the output port ?

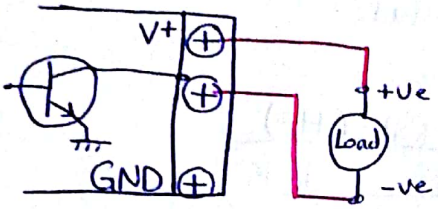
Q2



① What's the type of the signal source? Floating

② Connect the PV output to the DAQ?

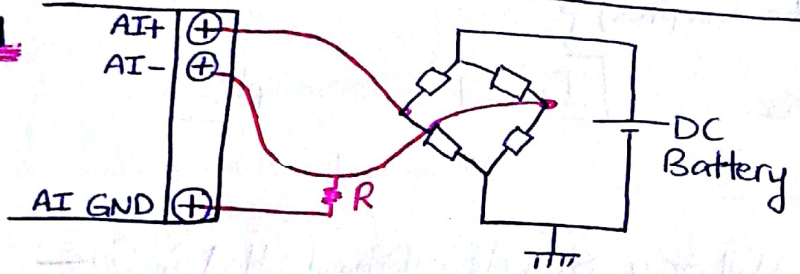
Q3



① What's the type of the digital output configuration? Sink

② Connect the load to the DAQ?

Q4



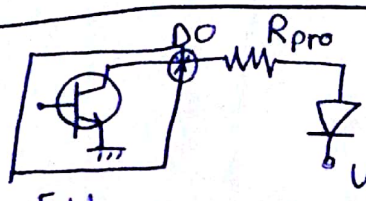
[floating sensor + differential DAQ]

① Connect the CCT output to the DAQ ?

② What's the type of the signal source? floating.
الي فالسنسر
DAQ

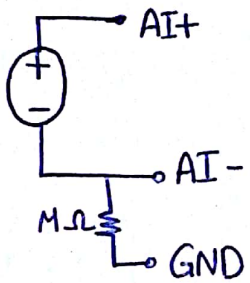
---

*



$$R_{pro} + R_{load\ (100\Omega)} = \frac{\Delta V}{I_{rated\ min}}$$

اذا طلع جوابها بالسالب يعني ما فيه protection



شو بصنف هاي الحوامل؟

a. Single Ended
b. Grounded
c. Floating

differentio

*Sample Rate (KS/s) →

N Samples	samples to read	Rate (HZ)
	10	5 HZ

(Sample) بدك 5 تقرأ كل هاد الـ Rate

وبدك 10 samples — 5 samples in one second

It will take **2 seconds** to collect all the samples

Q: For the port where the potentiometer is connected above, the following configurations have been made:

Timing Settings

Acquisition Mode	Samples to Read	Rate (HZ)
N Samples ▽	500	10K

* What's the rate of update of the program (how long would it take to collect all the samples)?

$$\frac{500}{10*10^3} = 0.05 \text{ sec.}$$

[I need 500 samples]

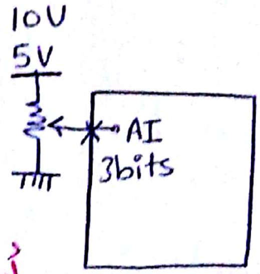
← بينما لو كنا طالبين Sample on Demand ، بيصير التأير الي بال Loop هو اللي بحكم
كل ما خلص التأير الي بالوب بجيب Sample وحدة

* Resolution →

analog input $Res = \frac{V_{in(max)}}{2^n - 1}$ → min. voltage change required to be detected by the DAQ

$R_{in} = \infty$ [high input impedance] → to reduce the loading effect

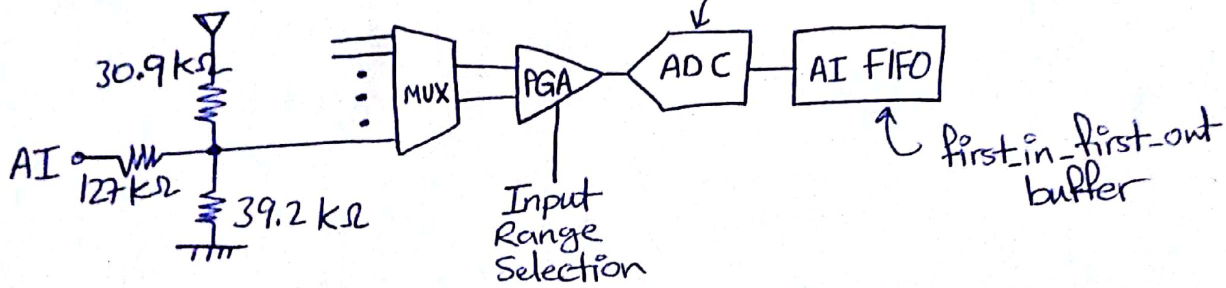
$R_{out} = \emptyset$ → عشان ما يصير بينها وبين اللود voltage division فبطل أقدر أعرف الفولتية اللي رايحة ع Load



أصغر $Res = \frac{5-0}{2^3-1} = 0.7\ V/bit$ → Res. لما أقل اذن الدقة أعلى

$Res = \frac{10-\emptyset}{2^3-1} = 1.4\ V/bit$

analog input circuitry :

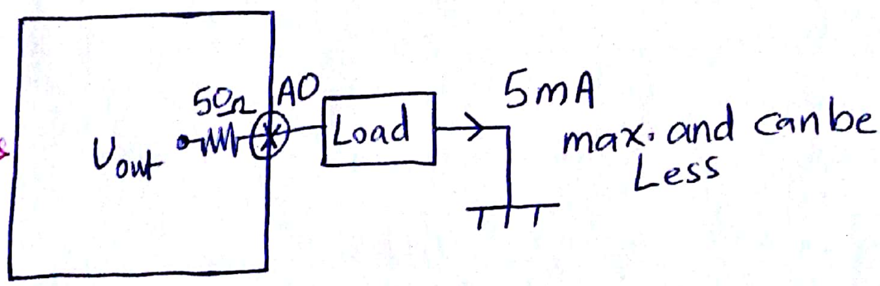


---

analog output $Res = \frac{V_{out(max)}}{2^n - 1}$

Each analog output has as on-board 50Ω resistor and can drive loa… with a max. current of 5mA ?

analog output calculations

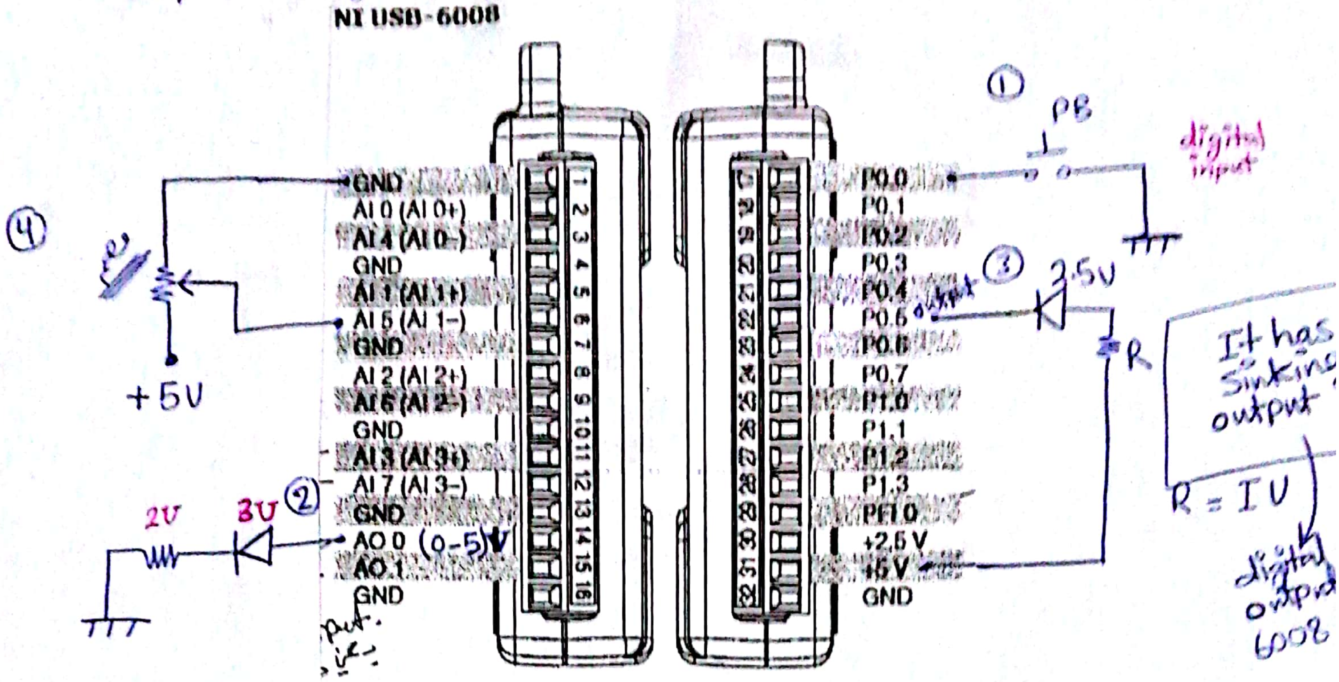


Scanned by CamScanner

Questions

Q1| Make the proper connection and calculations to perform the following task ?

1. A PB must be connected to the DAQ to read the no. of times it has been pressed. → input
2. A light intensity of a LED 3V connected to the DAQ must be controlled based on your program → output
3. A LED 2.5V connected to the DAQ and will be controlled to turn on and off.
4. The output voltage of a potentiometer is to be connected to the DAQ.



Q2| Suggest an input CCT that will match:

Voltage input setup

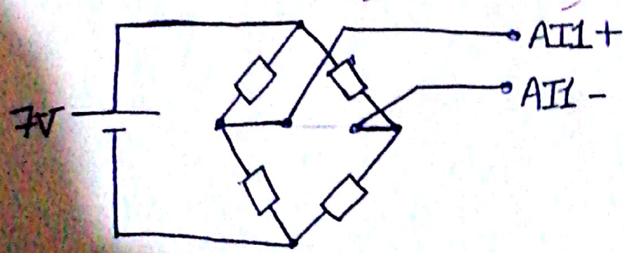
Signal Input Range		Terminal Configuration
Max 7 / Min 0	V	Differential

① بما إنه من (0-7) موصلين (0-5) يعني لا أحتاج لـ [external power supply]

② بما إنه في [external power supply] يعني شغالة على سنسور من نوع floating so single ended

③ نوع الـ DAQ ← diff.

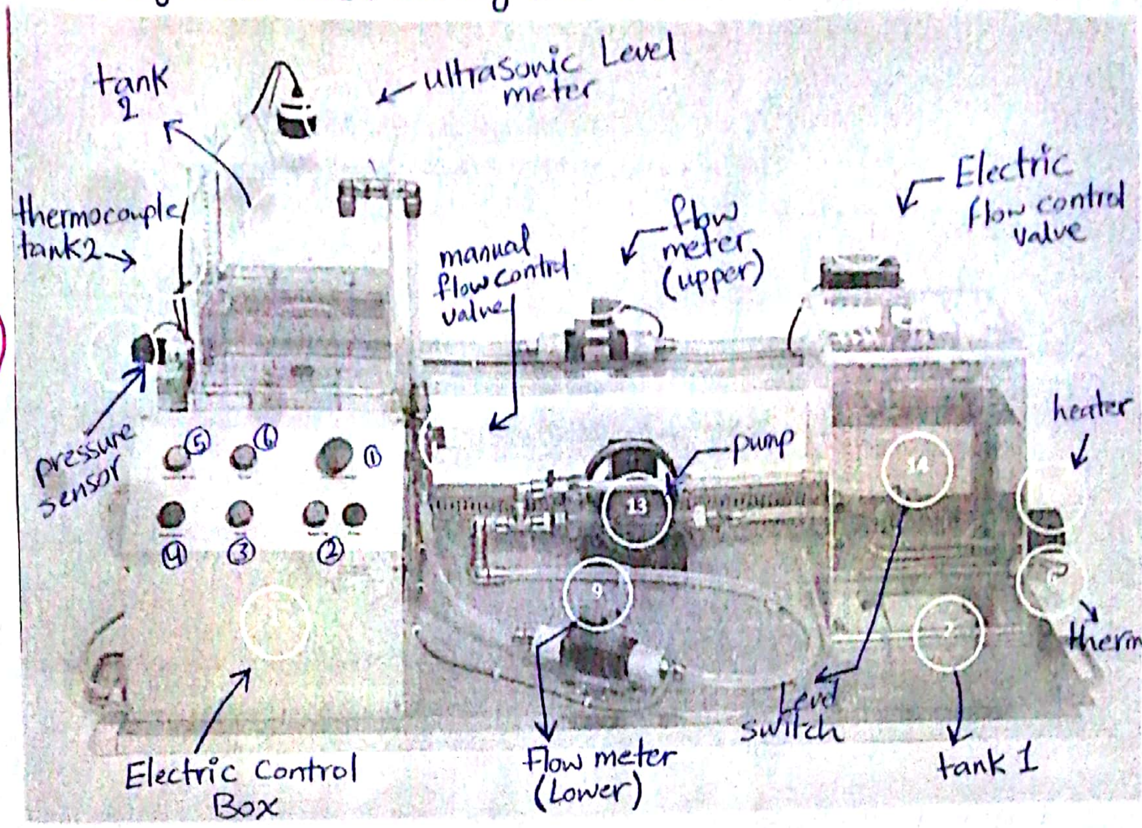
④ الفولتية موجال mV ليرتبط [illegible] منه TC لهيك أجاب WB



#

Exp(3): Process Trainer System

The system used during this experiment is the PTool system.



مهم جدا

- process trainer for → Level / → flow / → temperature / → pressure } control.

Level ≡ pressure

- In the electric control box shown in the figure above, it contains :-
  1. Emergency Switch
  2. main power
  3. Pump Indicator
  4. Level alarm
  5. Process Indicator
  6. Heater Indicator

pump → system Actuator

- To control the level of the water in (tank 2) → من خلال قراءات السنسورات

  اللي فوق هو اللي منتحكم فيه

  - *ultrasonic level meter

    max flow = 25 [0-30 cm مع انه الـ tank2 مدرج من الـ لكن هاد السنسور أول 5cm ما بقدر يشوفهم]

  - *pressure sensor (الموجود عالأطراف)

    $P = \rho g h$ موقعه ٤ سم من تحت . فأول 4cm ما بشوفهم

sensor الـ output منهم عبارة عن current

أسهل من ما ننقل فولتية اللي ممكن يصير عندك drop voltage و losses أثناء ذلك

منحتاج 2 نظام على analog Input

– To control the flow of the water at (tank 2) :-

* flow control valve represents a disturbance in the system [as a control sys.] and a resistance [as a hydraulic sys.]

* flow control valve / actuator + sensor } لكننا بعتبرها actuator للـ system ككل

analog output — analog ↙ أتحكم بفتحة الـ tube — ↓ بيقيس تقريبًا مدى فتحة الـ valve

* Input flow meter → system input

[سنسور بعرف منه معلومات عن الـ input]

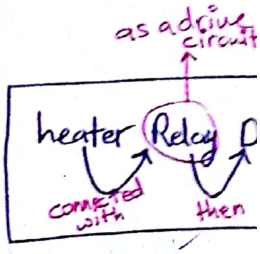
[Level control] ما بيعني الـ (input flow meter) إذا بدل

– Heater → موجود في أسفل الـ (tank 1) من تحت وهو عبارة عن مقاومة حرارية بتسخن وبتسخن اللي حواليها وإذا ما كان في مي رح تحرق حالها

digital output

عشان هيك في (Level switch) (digital input) لتفصل البور عن الـ drive CCT حتى يتعبى الخزان مي وتقدر تشغل الـ heater

مبدأ عمله (on/off) فهو غير قابل للتوصيل المباشر مع الـ (DAQ) فنحتاج لـ relay



– temperature sensors (thermocouples) → [موجود بداخل كل tank]
J type

الـ ↙ بيشغل الـ heater بناءً على قراءة الـ Level switch

Suitable for this process (accurate + Linear)

interfacing فنحتاج لـ voltage [mV] ⇐ (sensor) بعطي هاد الـ

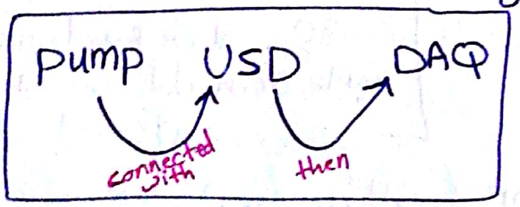
– Pump → هي الـ System Actuator

والـ output منها بيمثل الـ input للـ system

في الانفيرتر (Variable Speed Drive) للتحكم بالـ q (flow rate)

بتاخد سيجنال من الـ DAQ → لتعطيك معلومات عن السرعة (analog output)

(الـ pump بتاخذ 2 signals) → on/off للمضخة (digital output)



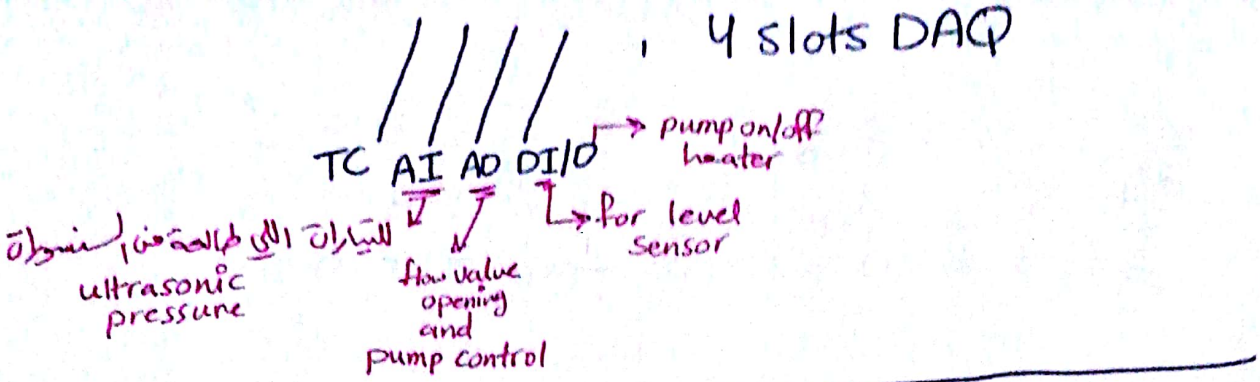
THE HASHEMITE UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF MECHATRONICS ENGINEERING

Mechatronics Systems Lab | Fall 2018/2019 | Eng.Shatha Al-Qadomi | Eng. Sarah Al-Barghouthi

Assignment 1: Process Trainer Kit addresses

Complete the following table in order to comprehend the idea of the data acquisition system with the inputs and output. Performing this assignment will make programming the DAQ much easier.

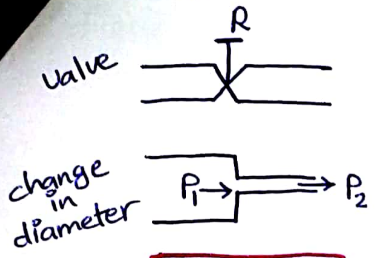
	Input/ Output (to/from DAQ)	Digital/Analog	Output Signal Type (voltage/current)	Module Number (بالبنوال)	Channel Number	Output Signal Range (فالبنوال)
Pump Power (analog power)	Output	Analog	Voltage	9263	0	0-10V
Upper Tank Temperature	Input	Analog	Voltage	9211	0	0-80 mV
Lower Tank Temperature	Input	Analog	Voltage	9211	1	0-80 mV
Pressure	Input	Analog	Current	9207	8	4-20 mV
Pump Enable	Output	Digital	Voltage	9472	0	
Level	Input	Analog	Current	9207	9	4-20 mV
Heater	Output	Digital	Voltage	9472	1	6-30 mV
Input Flow Rate	Input	Analog	Current	9207	11	4-20 mV
Output Flow Rate	Input	Analog	current	9207	10	4-20 mV
Flow Control Valve Feedback	Input	Analog	Voltage	9207	0	0-10 mV
Flow control Valve Opening	Output	Analog	Voltage	9263	1	0-10 mV

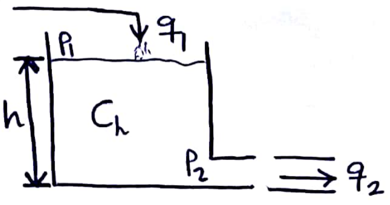


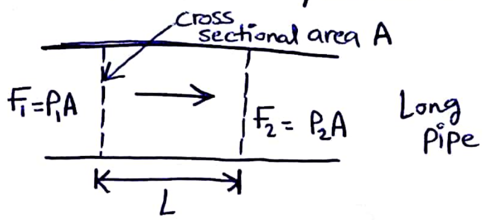
Questions

* Fluid System Modelling →

There are 3 basic building blocks : Resistance | Capacitance | Inertance







Resistance | **Capacitance** | **Inertance**

Resistance

$$P_1 - P_2 = R\,q$$

↓

It's the resistance to flow which occurs as a result of a liquid flowing through valves or changes in a pipe diameter.

ΔP يزداد (R_h) كلما زادت for a given rate of flow (q)

Capacitance

$$q_1 - q_2 = \frac{\partial v}{\partial t}$$

$$q_1 - q_2 = A\frac{\partial h}{\partial t}$$

assumed that the system is incompressible (ρ : constant) → $\left(q_1 - q_2 = \frac{A}{\rho g}\frac{\partial P}{\partial t}\right)$

$\frac{A}{\rho g}$ = C

↓

it describes energy storage with a liquid where it's stored in the form of potential energy.

Inertance

$$F_1 - F_2 = P_1 A - P_2 A$$

$$ma = A(P_1 - P_2)$$

$$m\frac{dv}{\partial t} = A(P_1 - P_2)$$

$$AL\rho\frac{\partial v}{\partial t} = A(\Delta P)$$

$$\boxed{\Delta P = I\frac{\partial q}{\partial t}}$$

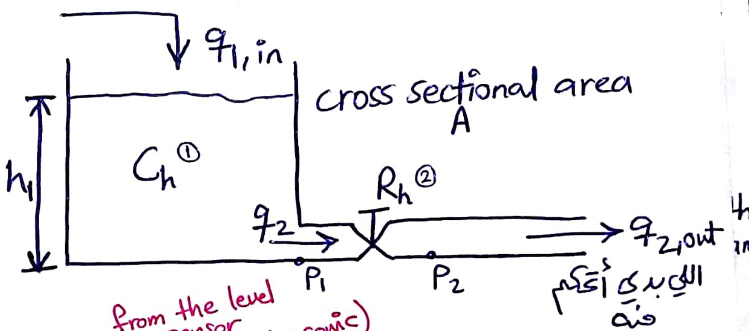
* محصلة القوى الكاملة causes the mass to accelerate with an acceleration (a)

↓

(I_h) equivalent of inductance (L) in electrical system, or a spring (k) in mechanical sys.

$$I_h = \frac{L\rho}{A}$$

⇒ For the process trainer system :



$$q_{in} - q_{out} = A\frac{dh}{\partial t} \quad — ①$$

[If $P_2 = 0$] $P_1 - P_2 = R_h\, q_{out} \rightarrow q_{out} = \frac{\rho g (h_1)}{R_h} \quad — ②$

h_1 ← from the level sensor (ultrasonic)

R_h → flow control valve

$$\boxed{q_{in} - \frac{\rho g h_1}{R_h} = A\frac{dh_1}{\partial t}}$$

[If $P_2 \neq 0$]

$$\boxed{q_{in} - \frac{\rho g (h_1 - h_2)}{R_h} = A\frac{d(h_1 - h_2)}{\partial t}}$$

PID Controller →

$$PID = K_p + K_d S + \frac{K_I}{S} = \frac{K_p S + K_d S^2 + K_I}{S}$$

$$PID = K_p e(t) + K_I \int e(t)dt + K_d \frac{de(t)}{dt}$$

P Controller → (reduce ess) بتقلل ess / بتزود oscillation ال

I Controller → (eliminate ess) / بتطيح / بتزود oscillation ال

ess	Σ ess
1	1
0.5	1.5
0.2	1.7
∅	1.7

← accumulation بتعمل

D controller → protective element, ما بستعملها لحالها أبداً

- highly sensetive to noise in the process variable signal
- causes the output to decrease if the process variable is increasing rapidly.

- بتقلل ال order system من 1st و 2nd، وهكذا
- ما إلها تأثير على ال steady state Respanse
- فقط بتقلل ال asc. بال Transient Respanse

*

Parameter	Rise Time	Overshoot	Settling time	steady state Error	Stability
K_p	dec.	inc.	small changes	dec.	degrade
K_I	dec.	inc.	inc.	eliminate	degrade
K_D	small changes	dec.	dec.	no effect	improve if K_d small

the process trainer system is a first order with a closed loop form
لكن بعد ما نعمله ديزاين عطريقتنا بيكون (second order) لتقليل oscillation ال PID

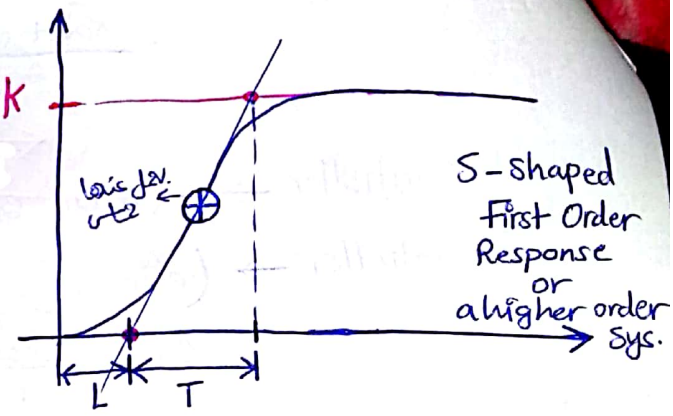
PID tuning using Ziegler and Nichols methods → لإيجاد قيمة $(K_P | K_I | K_D)$

[1] The First method:

$K_I = \frac{1}{T_i}$

type of Controller	K_P	T_i	T_d
P	T/L	∞	∅
PI	0.9T/L	L/0.3	∅
PID	1.2T/L	2L	0.5L

$T_d = K_d$



* PID controller unnecessary in this method because it's a first order system that sustains no oscillations, thus no need for the derivative action.

Open Loop

L: delay time

T: Time constant

K = y_{ss} (for a unit step input)

$$T(s) = \frac{K}{\tau s + 1}$$

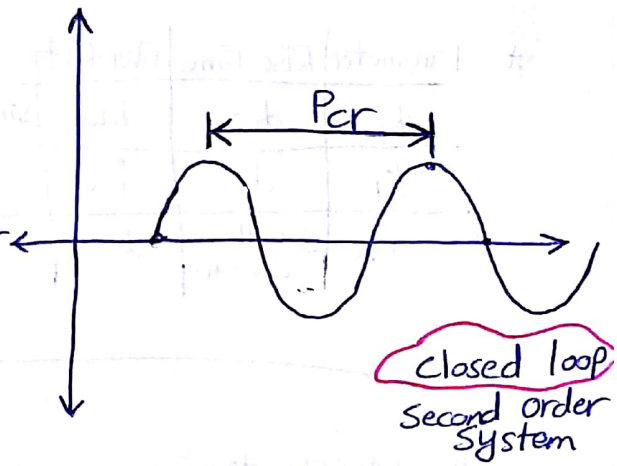
[2] The Second method:

using this tuning method for closed loop system, through only proportional feedback control to get pure oscillation

* Reduce the integrator and derivative gains to ∅ ← لغي الـ integrator و الـ derivative
* Increase K_P from ∅ to some critical value $K_P = K_{cr}$ which sustained oscillation occur. [if it doesn't occur then another method has to be applied]

K_{cr} هي قيمة الـ gain اللي بتخلي الـ osc. pure

وبستخدم K_{cr} و P_{cr} في الجدول



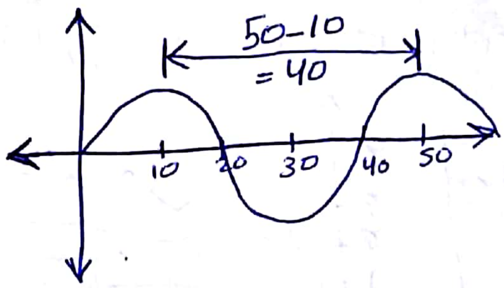
Type of controller	K_P	T_i	T_d
P	0.5 K_{cr}	∞	∅
PI	0.45 K_{cr}	1/1.2 P_{cr}	∅
PID	0.6 K_{cr}	0.5 P_{cr}	0.125 P_{cr}

$K_I = \frac{1}{T_i}$ $K_d = T_d$

[K_{cr}: هي قيمة الـ gain اللي خلتني أحصل على pure osc.]

Ex:

	Kp	Ti	Td
P	T/L	∞	∅
PI	0.9T/L	L/0.3	∅
PID	1.2T/L	2L	0.5L



	Kp	Ti	Td
P	0.5Kcr	∞	∅
PI	0.45Kcr	1/1.2Pcr	∅
PID	0.6Kcr	0.5Pcr	0.125 Pcr

Find PI transfer function?

بناءً على الشكل الحاصل (Oscillation) اذن أكيد ع الجدول الثاني (2nd method)

$PI = K_p + \frac{K_I}{s}$

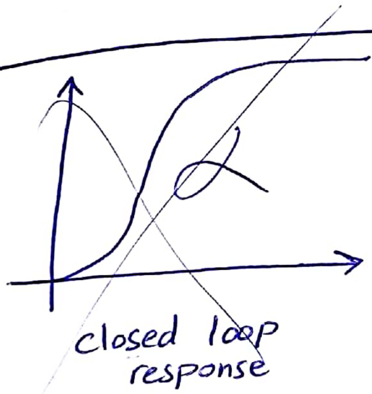
$T_i = \frac{1}{K_I}$

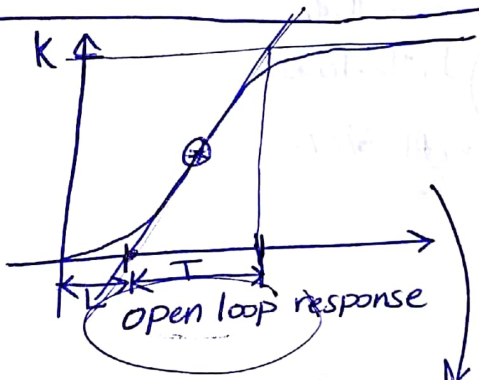
$T_d = K_d$

$P_{cr} = 40$

$K_{cr} = K_p$

Ex:





مناخذ بس ال open loop ومطبقين عليها

Type of Controller	Kp	Ti	Td
P	T/L	∞	∅
PI	0.9T/L	L/0.3	∅
PID	1.2T/L	2L	0.5 L

Find the PI controller transfer function?

$PI = K_p + \frac{K_I}{s}$

$= \frac{T}{L} + \frac{0.3}{Ls} = \frac{0.3 + (T)s}{(L)s}$

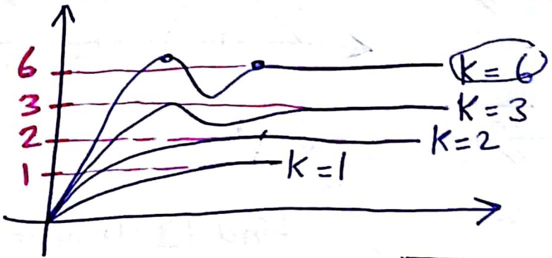
---

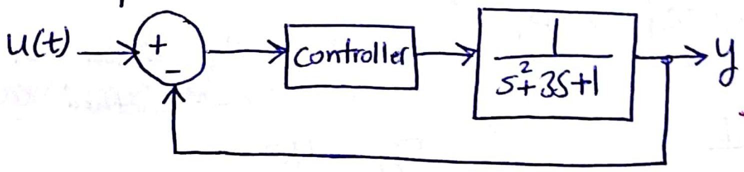
* If the input flow rate equals the output flow rate, then the tank must be empty (false)
* The pump is used to control the input flow to the upper tank (true)

Scanned by CamScanner

→ صمّم (PI controller) للنظام اذا علمت أنّ الرسم يمثل استجابة النظام لتغيرات المقابل K :
[Ziegler–Nichols]

لو كانت هاي ال response لـ closed ← السؤال ما بنحل لأنه لازم يكون عندي pure Osc. بدون ثبات





→ Closed Loop System

Second Method (pure Osc.) بيلزمني

$K_{cr} = 6$ ← [k = 6] بصمّم عند

وبطلّع (Kp) من الجدول

هاي الريسبونس لكن لو كانت لـ (open Loop) مناخذ الي فيها S-shaped