

 *Basic Programming for Electronics*

**WORKLOAD DISTRIBUTION**

|  |  |
| --- | --- |
| **Full name** | **Task** |
| Duong Doan Tung | o Coding software   * Edit video |
| Nguyen Trong Huy Hoang | * Setup hardware * Decorate |
| Le Hoang Nam | * Write report * Buld hardware |

W o r k l o a d d i s t r i b u t i o n

# TABLE OF CONTENTS

**SECTION 1: INTRODUCTION ................................................................................ 4**

**SECTION 2: PROJECTS DETAILS ........................................................................ 4**

1. **GENERAL INTRODUCTION** ...................................................................................... 4
   1. *Some links .......................................................................................................*  4
   2. *Idea ................................................................................................................... 5*
   3. *Software interface ............................................................................................. 5*

**SECTION 3: DISCUSSION ..................................................................................... 11**

**SECTION 4: REFERENCES ................................................................................... 12**

# Section 1: INTRODUCTION

***Abstract – Educators and researchers worldwide are using National Instruments products to automate routine tasks, accomplish new objectives, replace outdated and expensive equipment, and demonstrate students the potential of high technology. Engineers have used virtual instrumentation for more than 25 years to bring the power of flexible software and PC technology to test, control, and design applications making accurate analog and digital measurements from DC to 2.7 GHz.***

***The goal of this paper is to teach students basic concepts of LabVIEW programming, that can be used to easily integrate hardware and software to acquire, analyze, and present data. The block diagram of your application enables you to define operations to be performed on your data. The front panel allows the user to interact with a program while running.***

# Section 2: PROJECTS DETAILS

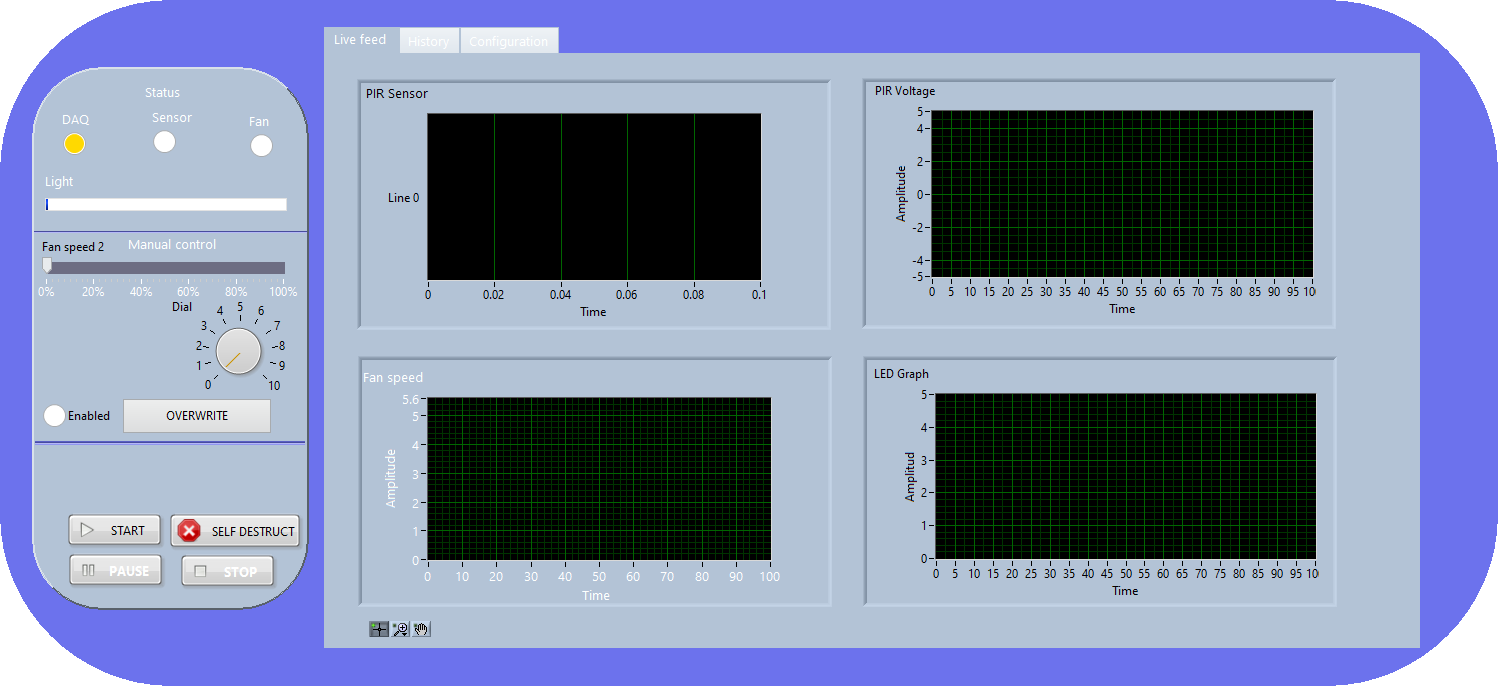
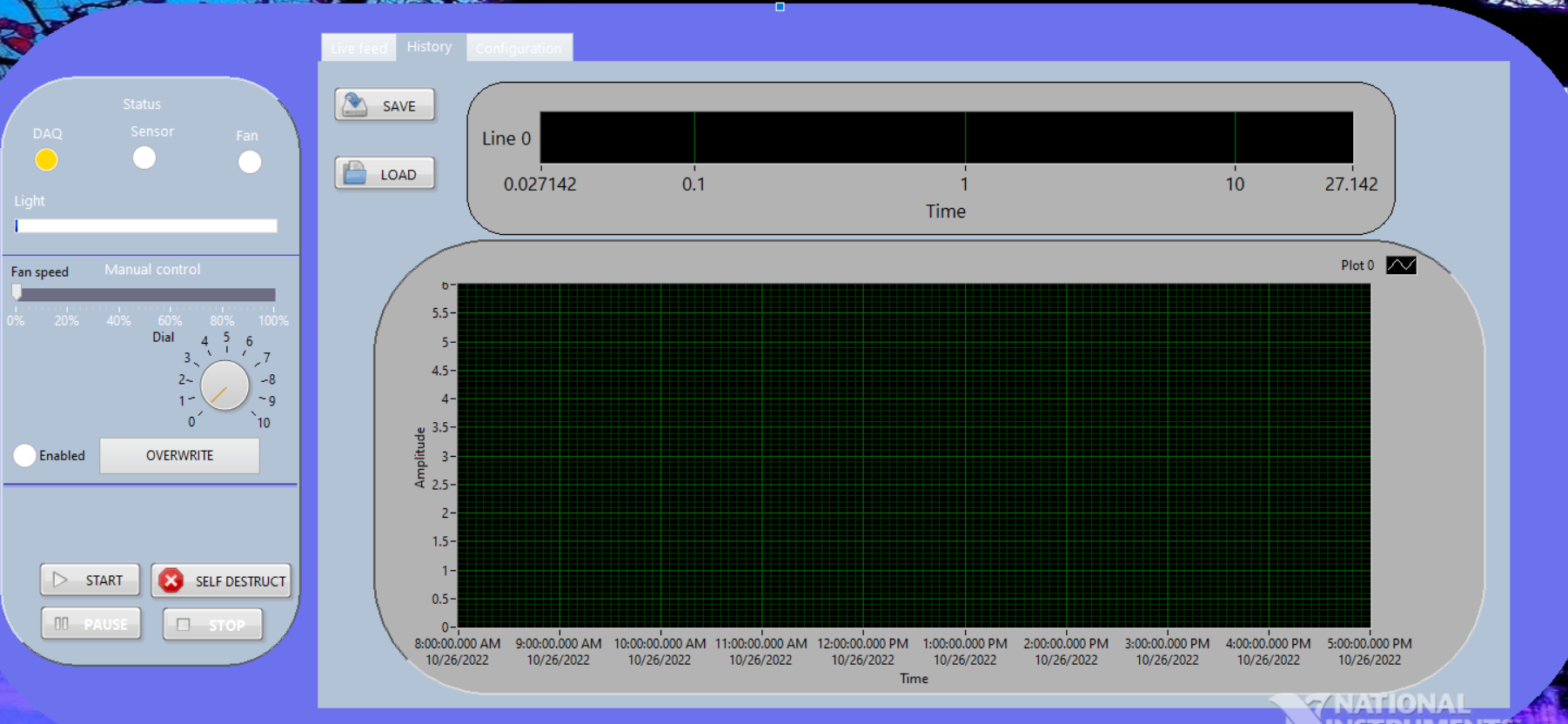
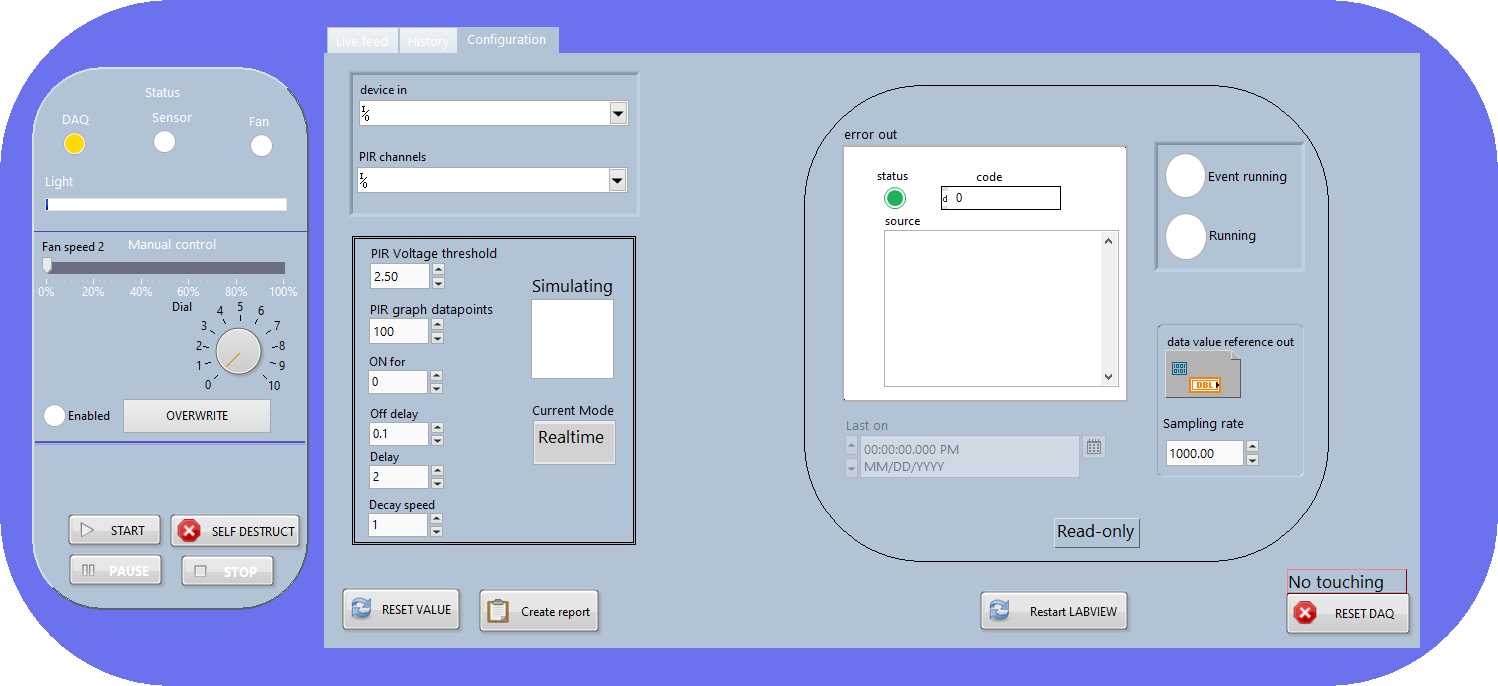
**1. General introduction**

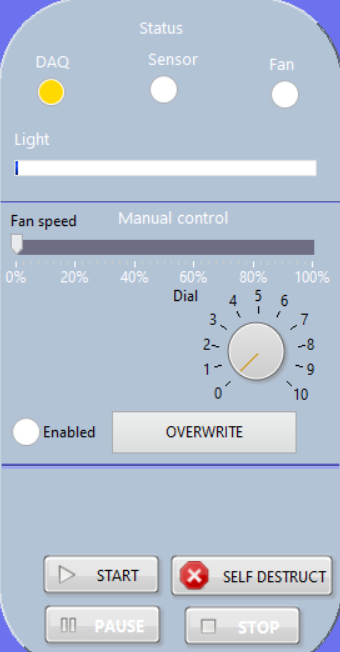
## 1.1. Some video links

Link to folder contain Presentation file and Demostration videos : <https://drive.google.com/drive/folders/15Gyvd-AxRdvlOK2FSbTaDAAlVZT9CAch?usp=sharing>

***1.2. Idea***

In this project, we carry out the topic assigned by the teacher, which is the control system and data collection, data processing PIR sensor to detect people to turn on/off lights or fans. LabVIEW software is used to communicate, collect data, and control via NI-DAQ.

***1.3. Software interface*** ******



First of all,this is the state table (control panel)

In which:

* DAQ has 3 colors:

-Yellow : Run simulation

-Green: successfully connected to DAQ

-Red: Failed to connect to DAQ

* SENSOR: Lights up when a set threshold is crossed
* FAN: Lights up when the fan is running.
* Light: Indicates the brightness of the lamp

We have 3 control mode in our project :

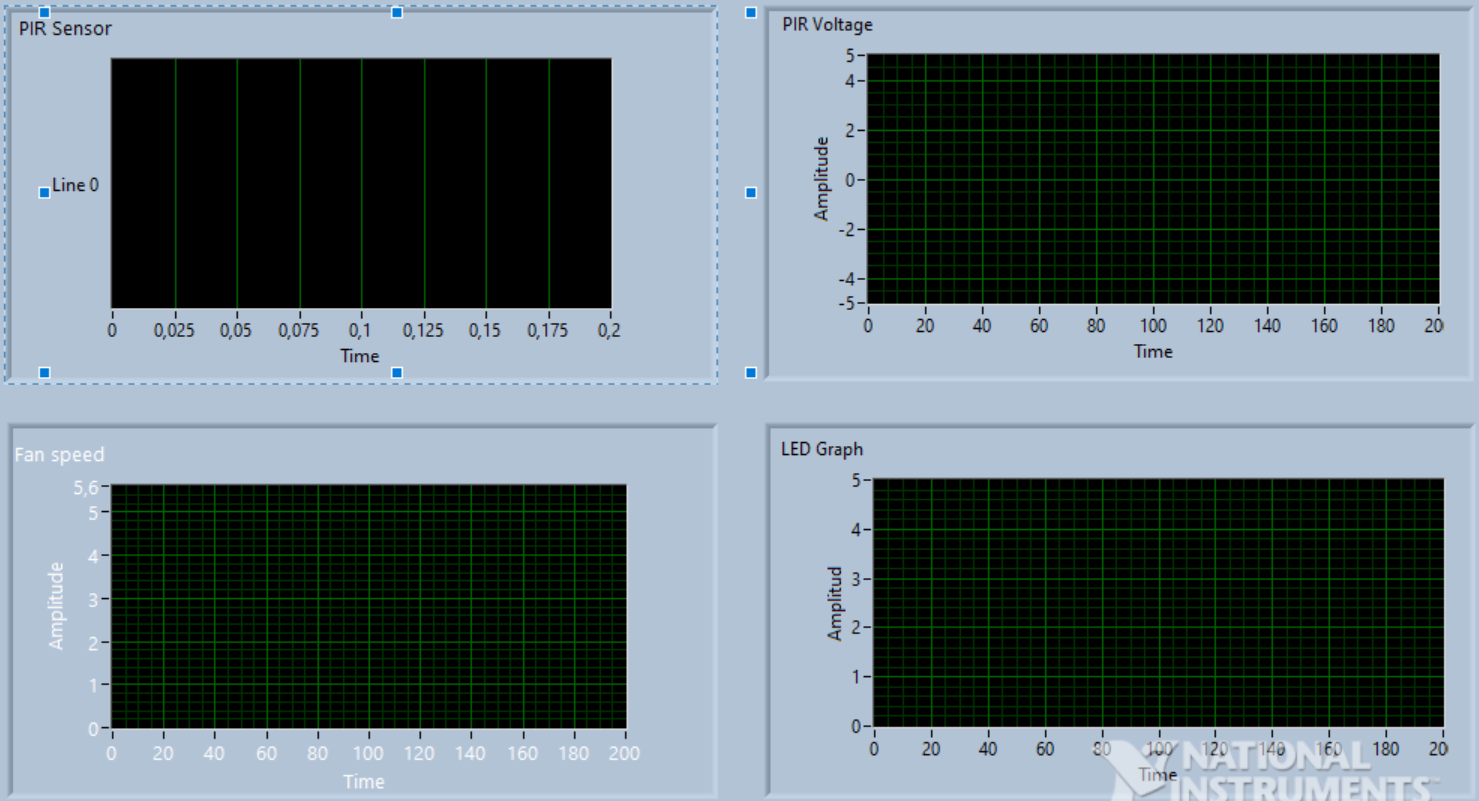
Schedule mode : create a schedule for lights and fans to work automatically, but due to lack of actual data, it is currently not possible.

Manual mode : allows users to proactively define project parameters.

Realtime mode : show us actual data over time.

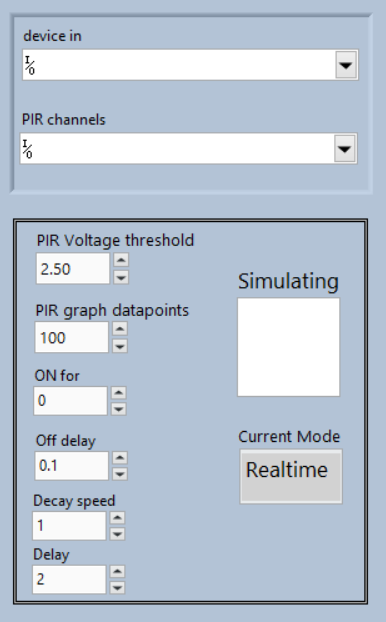
When the light is on, the ENABLED will light up (haven't done it yet, so don't know)

During execution, do not press SELF DESTRUCT.

TAB Live Feed 

In this tab show us PIR Sensor processed data, PIR Voltage sensor voltage, Fan speed displays the speed of your fan, LED Graph displays lamp brightness.

TAB CONFIGURATION



This tab use for to edit, change data parameters. In this tab we have:

* Upper frame: Select the port to connect to DAQ)
* Lower frame:

-Sensor rating (currently set to 2.5V)

-Amount of data displayed on the chart (100 data points)

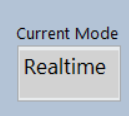
-ON for: the time after disconnecting for the light to turn off (for example, the value ON for = 1s, after 1 second, the Fanspeed and LEG graph will decrease to 0)

-Off delay will come with decay speed: if the off delay is 0.5 seconds and the decay speed is 1, then every 0.1 second after the light goes off, it will decrease by 1 from 5V until it reaches 0 ).

-(2 parameters used to control the slope of the fan and led graph)

 Read-only table: track errors and parameters!!!

Reset value: reset all the value on Front panel to the original dèault value.



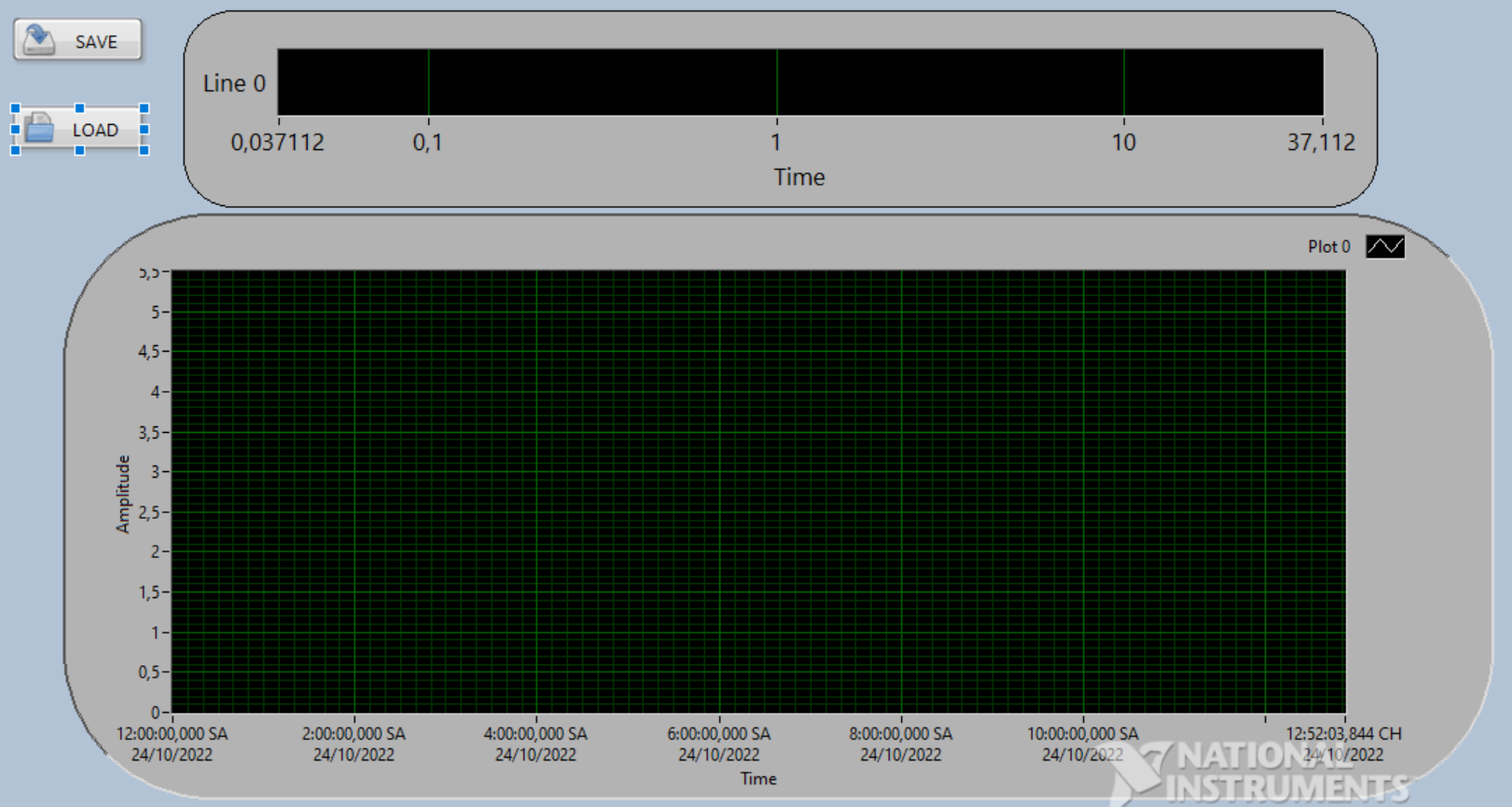
TAB HISTORY

Table above: Save the processed PIR SENSOR data.

Table below : Save the data of Light and Fan.

We can break the block diagram in to 2 part: Main program and a event handler

### **Main program**

The main program is Flat sequence structure. It contain 3 frame which execute in order from left to right

The first frame to set up all required variable each run. This frame will run each time the user press the Run button.

The second frame is to check DAQ connection and determine to run in simulation mode or not: It try to send a signal to turn on fan to the DAQ, the error out from DAQ write wired to simulation check to handle the error.

Last frame contain a while loop to read the sensor data continuously:

* The data from DAQ then sent to 3 independent section: Manipulate Output subVI, Show waveform on Live feed tab, save data on history tab.
* Manipulate subVi handle everything need to control light and fan.
* Set waveform subVi create and combine data to show as waveform on Live feed tab.

In the manipulate output subVI, we first compare the sensor data to the threshold. If it exceed the threshold, Last on variable will be set to current time and trigger the program to start output signal to Fan and LED. The user can control how long the output signal stay on, how fast it goes between on and off, and delay between Fan and LED.

The set data waveform subVI take 3 argument: The waveform before it, current data and datapoint value. and it will replace the input waveform with a new waveform with new data in it. It also output a waveform called subWaveform contain newest data to datapoint.

Beside all of the subVi above, We also implement 3 subVI: Error collector, Error handler and DAQ handler to collect and handle all possible error case happen.

# Section 3: DISCUSSION

Due to the limited implementation time, we lack actual data so there are still limitations. Therefore, to make it better, we need to do research deeper to apply more advanced knowledge... In the future, we will try to develop our project to apply in other environment and it also works good.

# Section 4: REFERENCES

1. <https://www.ni.com/docs/en-US/bundle/pci-pxi-usb-6255-specs/page/specs.html>
2. <https://www.ni.com/en-vn/support/model.usb-6255.html>
3. <https://forums.ni.com/t5/Example-Code/Programmatically-Reset-VI-Example/ta-p/3532614>
4. <https://forums.ni.com/t5/LabVIEW/Error-363507-LabVIEW-could-not-verify-the-authenticity-of-the/td-p/2881254>
5. <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjpm4a1g_76AhUrq1YBHYFkC70QFnoECBEQAQ&url=https%3A%2F%2Fwww.sparkfun.com%2Fdatasheets%2FRobotics%2FL298_H_Bridge.pdf&usg=AOvVaw0wIXqbF0WzC_SUFkv1WstB>