
Programming the Control Software of an Atomic Layer Deposition System

Written by:

Gustav Juhl Johansen (s134266)
William Bang Lomholdt (s134269)

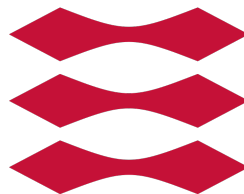
Main supervisor:

Matteo Todeschini, DTU DanChip

Co-supervisors:

Pernille Voss Larsen and Peter Windmann, DTU DanChip

DTU



Course responsible:

Erik Vilain Thomsen, DTU Nanotech

Report in course 33471 - "Nano-3W: Experimental Micro- and
Nanotechnology"

January 20, 2016

Technical University of Denmark

Contents

1	Introduction	1
2	Experimental Setup	2
2.1	Overview of Experimental Setup	2
2.2	Valves	2
2.3	Rough and steady state pump	3
2.4	Mass flow controller	4
3	Programming results	4
3.1	LabVIEW Modelling of the System	4
3.1.1	Main control	4
3.1.2	Pumps	6
3.1.3	MFCs	6
3.1.4	ALD subcycle recipe	6
3.1.5	Platinum and oxygen pulses	8
4	Discussion	9
5	Conclusion	10
	References	i
A	Solenoid valves	ii
B	Time calculation sub-VI	iii

1 Introduction

Depositing micro-, nano- and even atomic monolayers of materials in microscopes such as environmental transmission electron microscopes (ETEMs) can be an effective tool to investigate nanoscopic structures in-situ during addition of material.

However, equipment in ETEMs are expensive in comparison to external deposition setups, which is used in an atomic layer deposition (ALD) [1].

The issues in this subject are to optimize the system of atomic layer deposition, so the costs are minimal.

One of the issues, which is also the content of this report, involves programming of the software of an ALD setup. Using the National Instruments (NI) software Laboratory Virtual Instrument Engineering Workbench (LabVIEW), it is then possible to create a possible software to the ALD setup.

ALD works by injecting a gas or powder-cloud of material (*precursor*) into a chamber, where a substrate is placed, as shown in Figure 1.

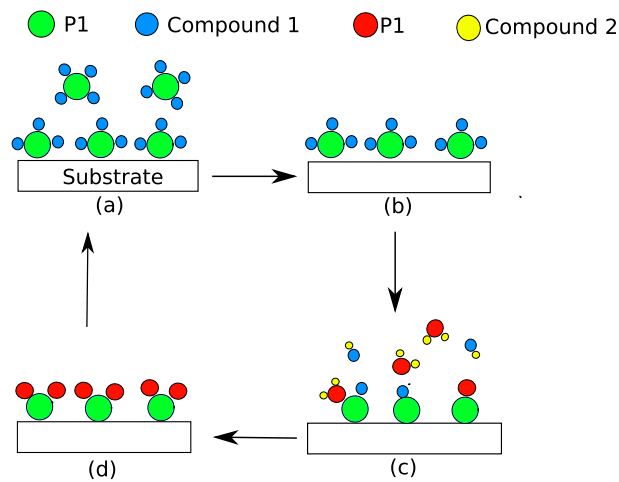


Figure 1: *Principal of ALD process. The process starts in (a) with precursor 1 a (P1) bounded to a compound, which interacts with the substrate. To remove excess molecules, purging is performed with an inert gas in (b). Process is repeated with new precursor (P2) in (c) and purging in (d), and then starts a new process from (a) [1].*

The precursors are deposited one by one on top of the substrate, with inert gas purging between each precursor switching, to ensure no excess precursor because of possible chemical reactions.

Compared to chemical or physical vapor deposition (CVD or PVD respectively), this deposition technique does not produce too many byproducts and hence gives a more uniform layer. Optimally the flow rate must be modelled as a step diagram as shown in Figure 2a, but in reality the step diagram looks as shown in Figure 2b.

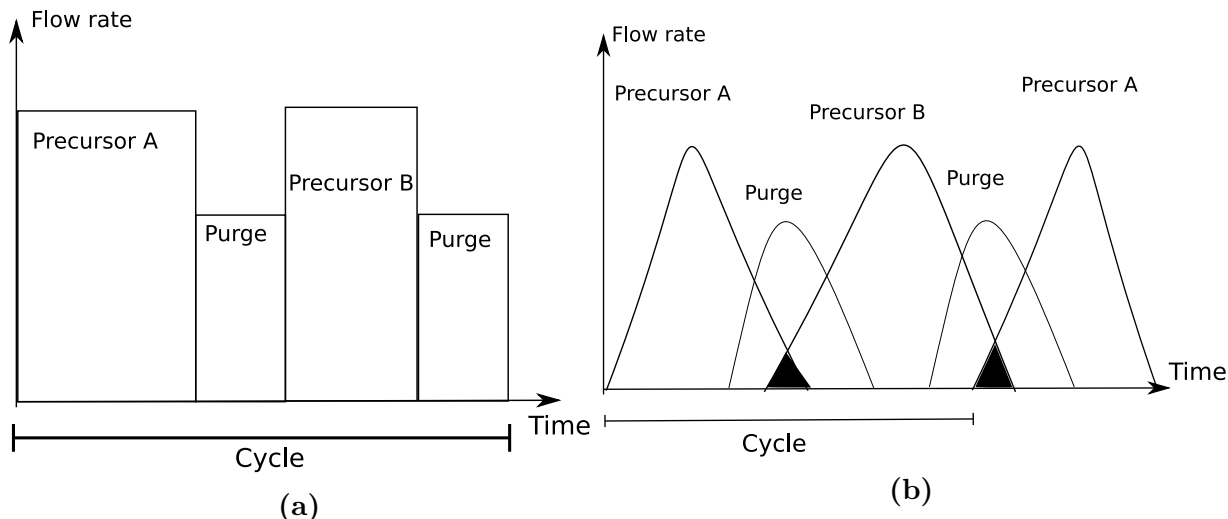


Figure 2: Cycle processes sketched as flow rate vs. time in the process. **(a)** The cycle process are optimal as box diagrams, which is the optimal way due to no overlap between the steps. **(b):** Actual process cycle, but is unfavorable, because the overlap (black areas) can result in CVD or PVD behavior.

2 Experimental Setup

2.1 Overview of Experimental Setup

A simple sketch of the system is shown in Figure 3a, while a picture of the system is shown in Figure 3b. The system consist of two pumps, two ALD valves where the precursors are injected into the system, and two mass flow controllers (MFC), which controls how much N_2 and O_2 is going into the chamber. By using the N_2 to clean out the system (*purge*) for other molecules, it is possible to make a thin uniform layer of a given precursor upon the substrate inside the chamber.

2.2 Valves

There are two ALD valves in the system, since two precursors are being deposited:

- Platinum as solid powder sublimed
- Oxygen as gas

In LabVIEW the ALD valves controls injection of precursors which means that they must open in a given time interval and then close again. The significant difference between ALD valves and solenoid valve is that an ALD valve is a solenoid valve with 3 ports resulting in controllable output direction, while the regular solenoid valves are either open or closed. Solenoid valves are described in Appendix A. The other valves than the ALD valves, shown in Figure 3a, are solenoid valves and one manual valve. The manual valve in the same figure is opened by the user, when the system starts.

For the two solenoid valves numbered ① and ② in Figure 3a, e.g. ① is open and ② is closed for rough pump mode, and vice versa for steady state pump mode.

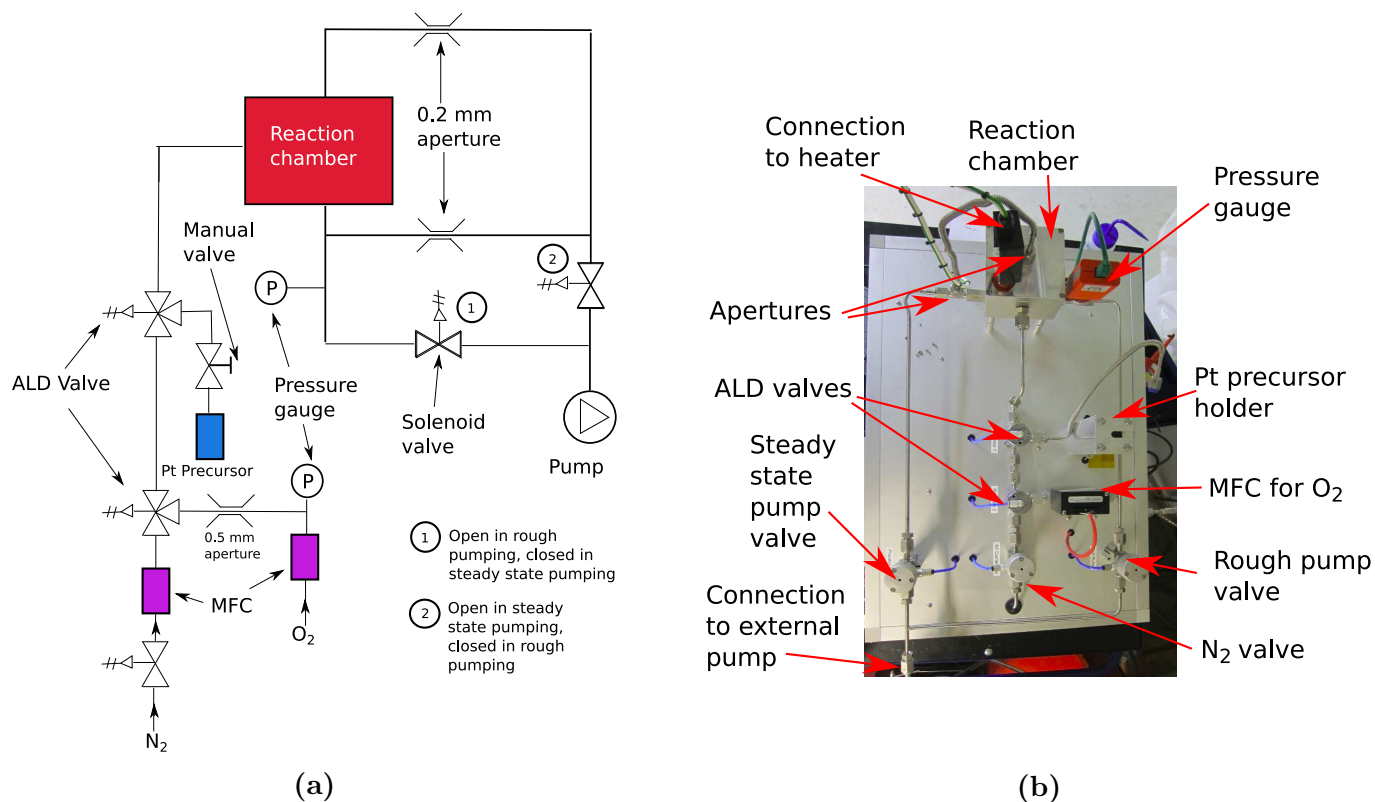


Figure 3: (a): Detailed diagram sketch of the system. All valves except for the manual valve and the ALD valves are basic solenoid valves (1 inlet and 1 outlet), described in Appendix A. Valve ① is open in rough pump mode and closed in steady state pump mode, and vice versa for ②. (b): Picture of the simple setup at DTU DanChip. The red arrows describe the components. A scroll pump is connected to the system externally, likewise the sources of the gases N₂ and O₂. MFC unit for N₂ is beneath the setup base. The rough pump and steady state pump valves are equal to ① and ② respectively in (a).

2.3 Rough and steady state pump

The rough pump drains the system for gas which decrease the pressure inside the chamber to a few mbar. While the steady state pump, as the name implies, keeps the system in a steady state and therefore keep the pressure constant inside the chamber. As seen in Figure 3a and Figure 3b there is actually only one pump, and in addition there is two solenoid valves which can "switch" between each pump mode and in that way control which pump is working.

In the system it is necessary for only one pump to work at a time, so the rough pump starts pumping and after reaching a specific pressure stops and switch to the steady state pump. This is also how it should be depicted in LabVIEW where the program starts after the steady state pump begins.

2.4 Mass flow controller

Mass flow controllers (MFCs) are used to control how much gas there is pumped through the system, and they are controlled by LabVIEW with three different inputs and/or outputs. First the MFC must be enabled by LabVIEW with a output, then a input is received from the feedback function in the MFC and finally another output is the demand function which controls the flow. Both the feedback and demand signals run in the voltage range between 0 V and 5 V.

3 Programming results

3.1 LabVIEW Modelling of the System

LabVIEW is used to control this system by having a NI USB-6341 connected to the separate parts in the system in one end and a computer in the other end by doing so a signal is going between the computer and the specific parts. With LabVIEW, it is then possible to control each pin in the USB and give "commands" to the system, e.g. opening a specific valve or making a sequence of tasks the system must perform. This is done in a logic system which is working by using true or false values, for example when the pressure in the chamber reach a certain value, the pump stops and the system is ready to work.

Basic terms in LabVIEW are Virtual Instruments (VIs), i.e. smaller program files and if the VI is very large, a sub-VI is created to make more space and give a better overview/control. The LabVIEW communicates with the NI USB-6341 unit through data acquisition (DAQ) software, a.k.a. "*DAQ assistant*".

3.1.1 Main control

The entire main-VI is shown in Figure 4, while its sub-VIs are presented in following sections. The most used term in the programming is Boolean Logic (true (T) or false (F) values), because the system must know if a condition is reached or not. Thus logic gates in the system are also used. Most commonly is the Selector connection, which has 3 inputs and one among two selectable outputs: one for receiving signal (Boolean), one value to use if the input is true and one if the input is false. Output is a value or argument, of which the selector has chosen to use in one of the specific cases. But also simple mathematical conditions and calculations, such as multiplication (\times) or less/greater than ($</>$) is used in the system.

In the main-VI, the system will start pumping when the user turns the "start pumping" button on. Then it uses the VI for the pump to control the pressure inside the system. Afterwards it reaches the entire deposition process where there is three sub-VIs, one for deposition, one for control of MFC for N_2 , and finally one that calculates the deposition time. The sub-VI for calculating the deposition time is shown in Appendix B. In the deposition process there is a emergency stop button and if pressed, the system stops the entire process.

After the deposition process is done, the user is given the choice to start a new process and if the user accept, the system makes it possible to change the input parameters and click ready when done and the process is repeated. If the user declines, the system asks if the user wants to vent and requests the user to loose the screws on the reaction chamber

lid. By accepting it stops the steady state pump and increase the pressure inside the system to 1 bar, and then stops the N₂ flow when the pressure is reached. If the user declines, the system stops and the process is done.

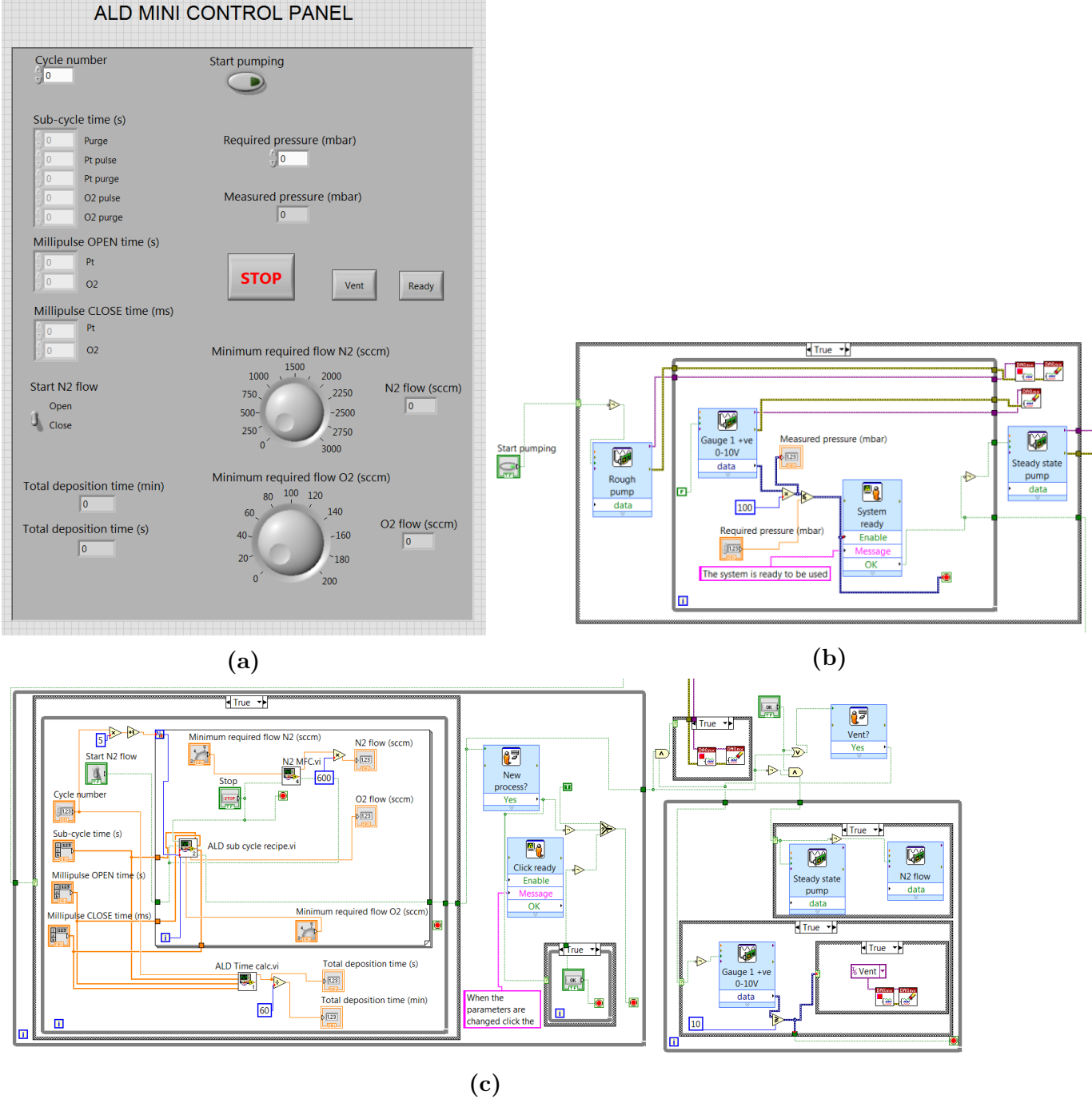


Figure 4: (a): System Control front panel, with input terminals such as knobs and type-lines, and also output terminals (b): System Control VI part 1. The broken lines continue on (c). This part contains the messages for rough pump and system readiness. (c): System Control VI part 2. This part contains calling the deposition process, the N₂ MFC and the time calculation.

3.1.2 Pumps

The rough pump is pumping inside the system until a certain pressure is reached. The rough pump then stops when the pressure in the system has reached the specified level, given by the user, e.g. 10 mbar. The VI is then programmed so it gets a T value when the rough pump is done and afterwards it switches over to the steady state pump and keeps it running. It then gives a output value of T or F which continues down to the deposition process. The VI of the pumps can be seen in Figure 4b.

3.1.3 MFCs

In LabVIEW, there is made two sub-VIs for the MFCs. These control the N_2 and O_2 flow. The MFC for N_2 is seen in the main-VI in Figure 4c and it begins by receiving a start input from the main-VI and then the MFC is enabled afterwards the flow inside the MFC is measured and shown, and if the flow is to low the MFC increases the flow, while keeping it constant if the required flow is reached. In this case the emergency stop button in the main-VI can also be seen here where it gives a steady state flow when it stops. Similar the O_2 flow is controlled where it opens when ordered and at the same time measures the flow of O_2 . The two MFCs can be seen in Figure 5.

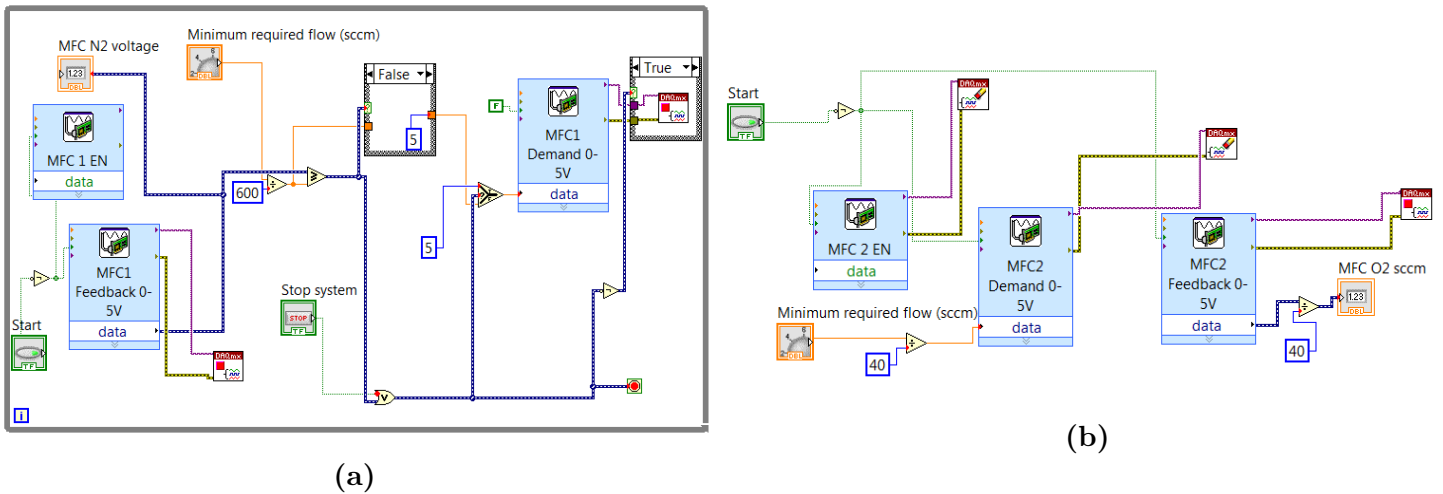


Figure 5: MFC sub-VIs where (a) is the MFC for N_2 and (b) is the MFC for O_2

3.1.4 ALD subcycle recipe

The ALD subcycle recipe is shown in Figure 6 and Figure 7. The process begin when it receives a T value and many other different user inputs which is used in the different cases. The structure of the sub-VI has been made so there is six different cases, which is a full cycle. The cases is specified by the pulse times and sub cycle times. The case has been made so it runs from 1-5 for a cycle and repeat this process if the user wants two cycles instead. When it is done it switches to the sixth case (case 0) where the system tells the user it has finished the process. In case 1,3 and 5 there occurs a purge at a user specified amount of time given by the sub cycle time input. In case 2 a platinum pulse occurs where the process consist of many small pulses and stops after a certain time. Case 4 is similar to case 2, but it now consist of a MFC for O_2 flow which opens when the O_2 should start flowing to the ALD valve.

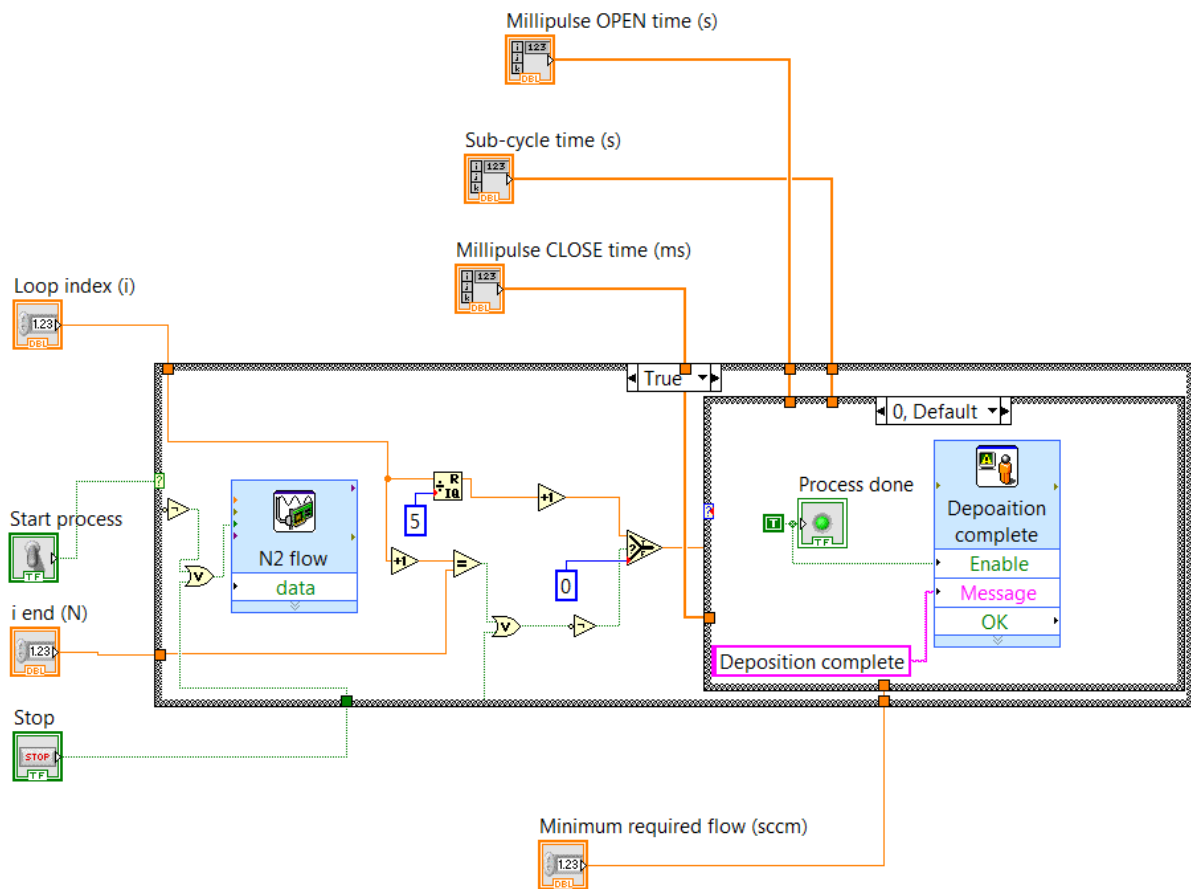


Figure 6: The ALD subcycle recipe sub-VI, with its default mode ("Process done").

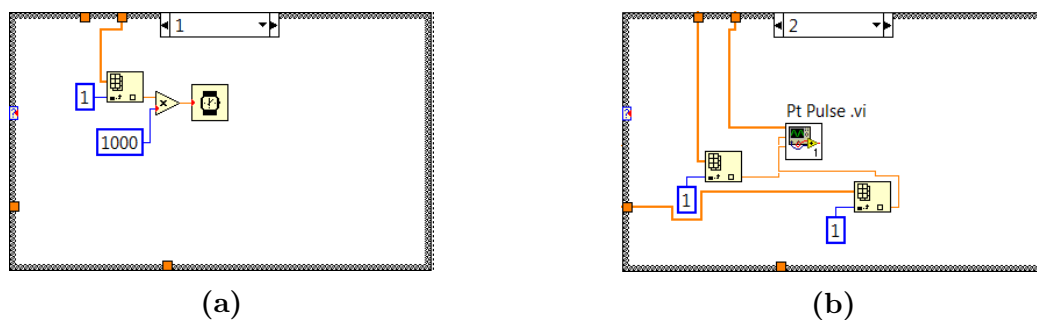


Figure 7: Other cases for the ALD subcycle recipe, case 1-2 is shown in the figure. **(a):** The first case where the system waits, i.e. it is purging. Similar for case 3 and 5. **(b):** The second case, starting the Pt pulse by calling its sub-VI. Same structure for O₂ (case 4), but with a MFC connected to it.

3.1.5 Platinum and oxygen pulses

For the platinum and oxygen pulses, it is necessary to have a pulse time which consist of many small pulses. A small pulse is a pulse where the ALD valve is open in 100 ms and then closed in 40 ms, and the process is repeated until a given time is reached. Their sub-VI is nearly identical, so in Figure 8a the number at the top, which is two, is replaced by four for the case of O₂. In Figure 8a is the case where the pulse is down, i.e. the ALD valve is closed, while in Figure 8b the case for top pulse is shown, i.e. the ALD valve is open. The clock in these two figures is a delay function, which make it possible for the user to control the width of the pulse top and down times.

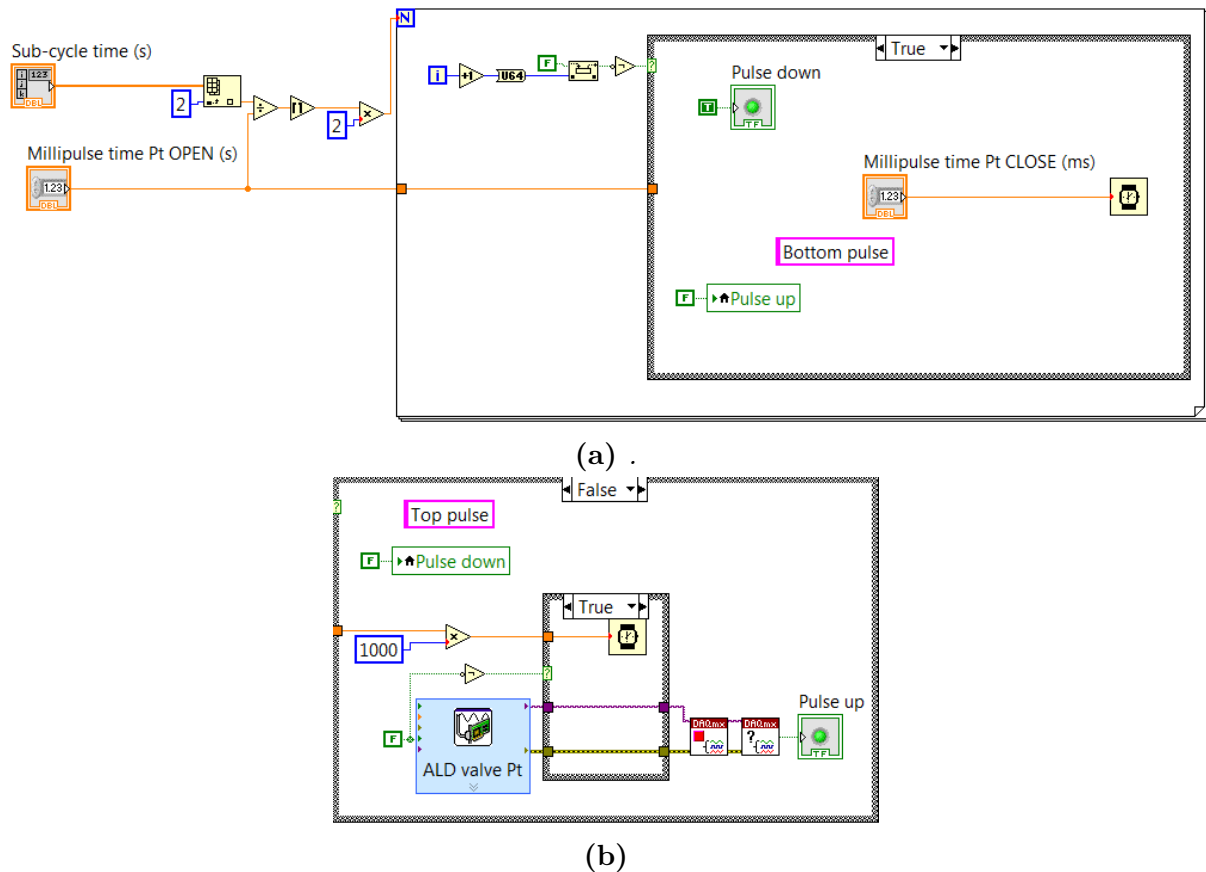


Figure 8: Pulse sub-VI, example for Pt precursor. The program for O₂ is similar. (a): This is shown with its true case. (b): This is the false case.

4 Discussion

In the program, which has not been tested due to lack of time, there is a couple of errors such as response time and the possibility that the ALD process makes a CVD or PVD process mentioned in the introduction and seen on Figure 2b on page 2. The program cannot be tested until a later date, but there are still possibilities to re-program the system.

First of all, the system can be done in a simpler way in order to reduce the response, computation and working time.

One possible way to test the software reaction/response time is to make a similar program in MathWorks MATrix LABoratory (MATLAB) program. NI suggest compatible solutions, e.g. reading data [2] and communication with modules [3]. If more advanced calculations than LabVIEW is able to do has to be performed, MATLAB has the ability for this.

Possible suggestions to check the waiting time are to make a MATLAB code with the exact same program as an algorithm, e.g. using a `tic-toc` command environment. Parameters such as time per sub-program or average time for the entire system could be conducted from this command.

Secondly, the parameters set as constants could have a knob or input terminal in order to control this. If you e.g. want a lower or higher required temperature, it will be easy to set it in e.g. the Pt precursor holder.

The NI USB-6341 is an excellent multipurpose unit in regard to setup software-to-experiment-connection, although it is more expensive than e.g. Arduino units [4,5]. However, sources of error during experiments could come for the unit, if it e.g. has too slow response time in comparison to the process flow steps. For some processes, even micro- or nanosecond time scales for the system reaction time are large amounts of time compared to e.g. the reactions in the chamber. Specific ALD equipment for use with LabVIEW can be required, e.g. a unit with sub-units for each MFC, valve and pump used. But this is one of the few units to be used with this system. The program could also be done so it is compatible to n precursors, e.g. programming a building block that expands while receiving an input for the amount of precursors, the user wants to use in the ALD process, as it is described in an article by Selvaraj and Takoudis (2015) [6]. However, the system has to recognize if the precursor is either in gas, liquid or solid state, and this could be programmed with case-structures.

LabVIEW itself is a very usable program for creating programs for specific purposes. When learned, it is easy to make the program you want. If your software is too large to make a program for using MATLAB, it is easier and more comfortable to draw lines among the building blocks than writing a code. However, the LabVIEW program is in comparison to written code based programming very limited. Unless you are a skilled software developer, you cannot program your own building blocks, and sometimes it is difficult to get packages for the setup you want a program for.

Physical tests of the system could be test on a sample with either low or high aspect ratio, and then compare it to the more advanced model from DTU DanChip.

A scanning electron microscope (SEM) or transmission electron microscope (TEM) image could then be conducted to see if the deposition gives qualitative and quantitative results. Thus it is important to use the exact parameters in each unit to get comparable results.

5 Conclusion

A possible solution to the software of the ALD setup has been programmed.

The program is created in LabVIEW with one main-VI and in total six sub-VIs. During programming, the Boolean logic system, numeric connection and simple mathematics have been used, along with DAQ assistant modules.

Since the program has not been tested at the physical setup, the programs may still contains some errors and limitations in respect to the ALD process and usability.

The software has been configured to control the pumps, MFCs, precursor pulses and purges, having the possibility of making a new process and venting the system in order to collect your sample.

However, there is still room for improvements, e.g. making a control for temperature, considering the need to control the temperature for different precursors in either powder or liquid state. Other future possibilities for the setup is to make deposition processes on substrate in the ALD, compared to the more expensive alternative e.g. the ETEM.

References

- [1] S. M. George, “Atomic Layer Deposition : An Overview,” *Chem. Rev.*, vol. 110, pp. 111–131, 2010.
- [2] “MATLAB DataPlugin info-site.” URL: <http://www.ni.com/example/29178/en/>. Last visited January 17, 2016.
- [3] “MATLAB USB-connection info-site.” URL: <http://se.mathworks.com/products/instrument/supported/usb.html>. Last visited January 19, 2016.
- [4] “NI-6341 retail price.” URL: <http://sine.ni.com/nips/cds/view/p/lang/da/nid/209069>. Last visited January 19, 2016.
- [5] “Arduino info-site.” URL: <https://www.arduino.cc/>. Last visited January 19, 2016.
- [6] S. K. Selvaraj and C. G. Takoudis, “Scalable control program for multiprecursor flow-type atomic layer deposition system,” *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, vol. 33, no. 1, p. 013201, 2015.
- [7] E. Bhuvanswarri, S. Anusha, R. A. Rukshna, and T. Devasheena, “Operating Solenoid Valve with NI-myRIO using labVIEW,” *International Journal for Scientific Research & Development*, vol. 3, pp. 354–357, 2015.
- [8] “Solenoid valve (Wikipedia).” URL: https://en.wikipedia.org/wiki/Solenoid_valve. Last visited January 16, 2016.

A Solenoid valves

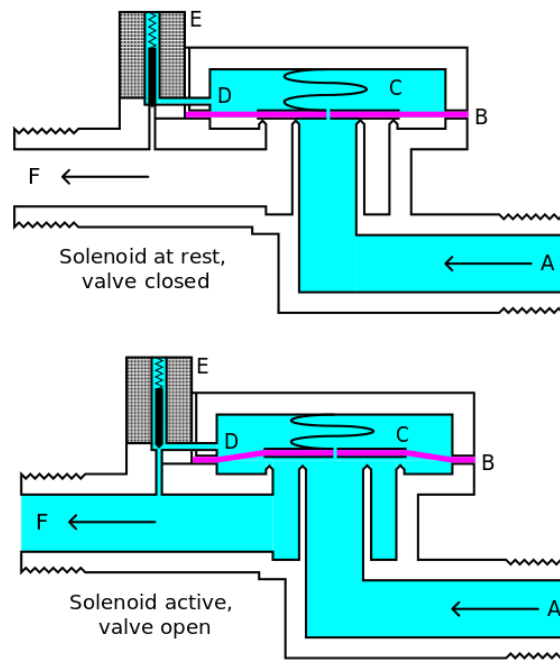


Figure 9: *The construction of the solenoid valve. It consist of the following parts: A: Input side. B: Diaphragm. C: Pressure Chamber. D: Pressure Relief Passage. E: Solenoid Unit. F: Output. When solenoid is active and thus the valve is open, the pressure in C is lower than for the A inlet, opening the valve [7]. Figure is adapted from [8]*

B Time calculation sub-VI

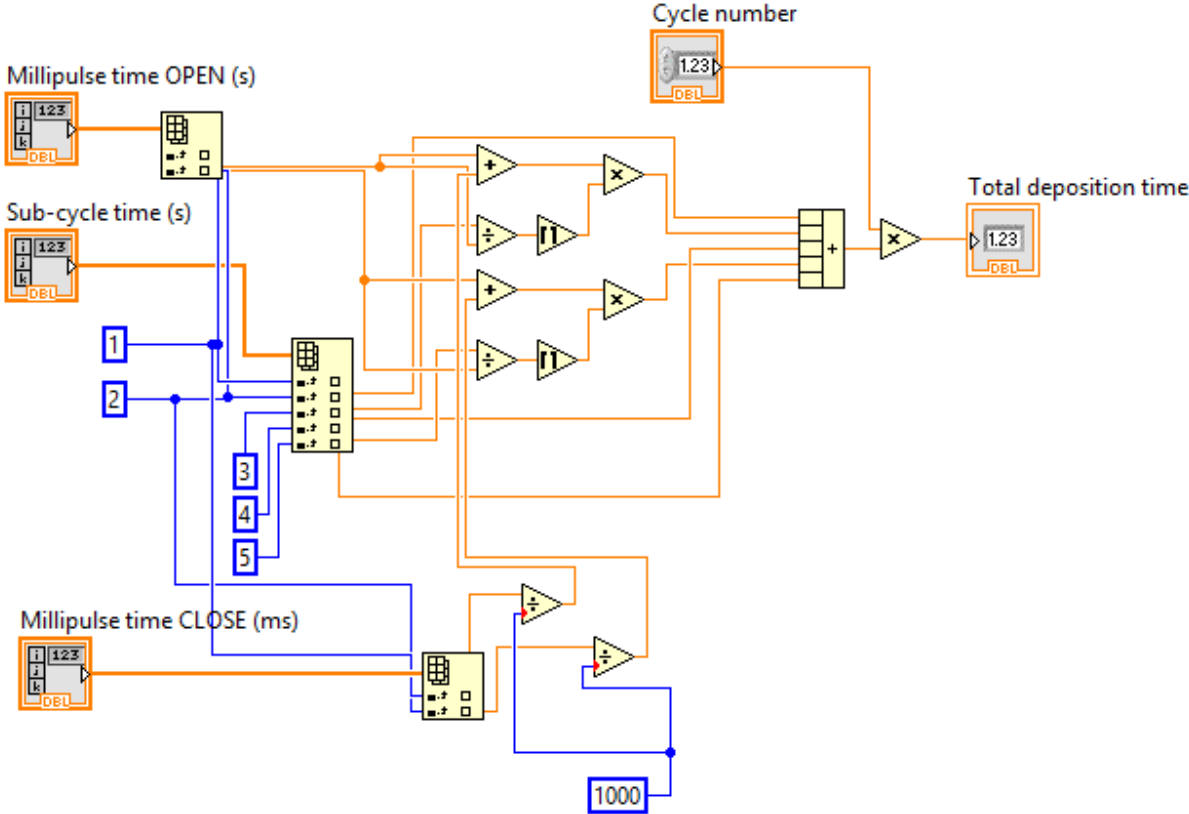


Figure 10: Time calculation Sub-VI, to calculate the time of the deposition time. Sub-cycle time is for e.g. pulse or purge. The upper millipulse time input terminal is for when the pulse is OPEN, and the lower is for CLOSE. Together they describe a period of deposition which takes a given time defined by the sub-cycle time. The cycle number tells how many times the process is repeated.