Progress Report for: #38693 - Maurice Petallo EIT

Period beginning: **Jan 4, 2016** and ending: **Jul 31, 2017**. **(19 months)**

Submission Date:	Aug 18, 2017
Supervisor:	Darcy Cook P.Eng. (#25060), Submitted on Dec 28, 2017
Mentor:	Jorge Viramontes Perez P.Eng. (#35283), Submitted on Dec 22, 2017
Period Employer:	JCA Industries, Inc
Job Title:	Software Developer

---

1. Give a description of the Engineering work experience you have obtained during this reporting period. Include information supporting the rest of your answers.

1 Overview of Work Experience

Within the period covered in this review, I worked as a software developer at JCA Electronics. In the engineering group, my task is to design electronic control systems application software for agriculture implements. I've been involved in three main projects that implements different farm application and solutions.

Part of my job is to review the system requirements and modify it as necessary depending on the implementation limits or initial design implementation results, improve the initial design to incorporate additional features per customer requests, implement the code develop the software as per design, continuously develop the new features, test the software implementation, create test documentation, create release documentation, and maintain the software for future enhancements. In addition, I was also tasked to engage with customer while they do their own testing. I do a lot of customer support both written and verbal. Apart from dealing with all those assigned projects and tasks, I spent efforts to improve our engineering software libraries and common software codes. I continuously contribute to our engineering knowledge base. I also extend my knowledge and experience to some of the new developers in the team. There are also times when I do minor fixes to old projects.

2 Application of Theory

2.1 Project 1

My first project is an embedded controller unit for a grain cart. The equipment has a very large bin that allows the harvester to do continuous operations in the field without having the need to load and unload the harvested grains from time to time. This bin is equipped with different features of controls allowing a farmer to simplify his work in the field.

2.1.1 Analysis

The project requirements document defines the control unit components. It also details out the algorithm and calculations that the controller has to do. To start with, I've done careful review of the requirements document. The document specifies a lot of formulas necessary for different calculations that the controller application has to do. I had to verify each of the formula. I had to actually do manual calculations and plug-in different numbers to make sure that it converts to proper measurements. I also calculated the extreme values that the can possibly take and make sure that it will be properly handled in the code for later implementation. Each of these formulas has to be checked and make sure that it will not give values that the system won't be able to handle.

2.1.2 Design And Synthesis

Based on the requirements document, I structured the entire controller application software into four major components namely: input, output, control and communication block. Each of these blocks are further subdivided into smaller sub modules which corresponds to different components and functions that the controller application will do, all of which satisfies an item in the requirements document. In my software code, I built each component one by one and then and connected the input and the output modules completing one functionality at a time.

There are other parts of the system where I put more focus such as the J1939 joystick device. The joystick module allows the controller application to detect when the operator of the machine has pressed a certain button. I had to review the joystick communication interface and make sure that the controller application will be able to communicate and command the system as per the intended design. Prior to this, I had to read the J1939 standard and make sure to take note of the important parts that needs to be considered as I implement the joystick message decoder at the controller application side.

Another design focus is the auger control movement using the proportional integral derivative (PID) controller. I've used a basic algorithm for this. With the guidance of our application engineer, I gathered unit step response of the actual system by driving the proportional valve leading to the auger with a fixed duty cycle. I simulated the behavior using our existing tools. I also implemented a simple PID block using our tools then I started to run different simulations. In our simulations, I selected different P, I, and D parameters and collected the data. Finally, I arrived with a range of values for our parameters for the simulation. I then used this information and quickly found the proper PID parameter settings as the customer tested on their end.

2.1.3 Testing

Before any development is completed, unit and system level tests has to be conducted to verify the design and functionality of the system. For this purpose, a test jig unit is provided. This unit is configured and set to vary different sensor input conditions. Using this, I verified each and every control of the system given a certain input condition. Using the light emitting diode (LED) components of the test jig, I was able to inspect whether output lines are driven properly. All possible bench simulations are done before deploying the software to the actual machine at the customer site.

Prior to the 1st controller application software release is the harness verification. It is also part of my responsibility to verify the correctness of the wire harness. A simple continuity and IO mapping verification is done to make sure that the lines are properly mapped, correct sensor signal levels are read in the controller unit end, and the output lines are properly driven. Actual controller units are also used for this test phase. Power up tests are also conducted with the 1st software release. This is done at the customer end. It is important to make sure that the system starts at the idle state for operators safety. The controllers unit has LEDs on-board to test it's aliveness. The more complicated functions are initially tested on the first customer visit.

2.1.4 Implementation

I started implementing the input module. There are a lot of input components in the system. These are mainly user controls such as a J1939 joystick device, different sensor devices such as angle sensors, speed sensors, proximity sensors, and transducers such as load cells. The controller box is equipped with analog-to-digital (ADC) devices and core drivers that takes care of providing digital voltage, current and frequency readings that can be utilized in the controller application layer. The input module takes care of converting these signals to angle readings in degrees, speed readings in radians per minute, open and close states, opening percentages, and weight in kilograms or pounds.

The heart of the controller system is the controller box. This is where the controller application runs. I started the development by reviewing the requirements document and made sure that each of the requirement answered all the customer needs and that all logic and algorithms can be implemented. To implement the design, I divided the controller application into four major blocks namely input, output, system control, and communication.

The output lines essentially drives two kinds of valves namely an ON/OFF valves or proportional valves. For the proportional valves, the output line going to the valve should be driven with a PWM signal to be able to move the component. I carefully checked that the output lines driving the valves will be driven in the frequency that is within the valve specification. As part of the output module, I designed the controller application with error protection logic that will allow the system to safely deal with these external device failures in the event of short circuit, open circuit, and over-voltage.

The control module is basically the artificial intelligence of the system. The core of the implementation starts in the state machine. This is what connects the input to output control. Except for direct on/off output, all the output tied to a control is managed within a state machine. I used this design to make sure that every output control is handled properly on different conditions. I created a number of state machine for different controlled output. The controller application has to process all the input signals and user controls and actuates control lines as necessary. This is also where the Real-time Operating System (RTOS) becomes a need for the system design. The control decision logic has to be executed quickly so the system would be allowed to respond at the right time. In the control module implementation, I’ve used mechanisms such as the proportional-integral-derivative (PID) controller. In this project, I’ve used a simple PID that will drive the proportional valves which essentially moves the hydraulically powered augers. I carefully created the system control not just to follow the design intent but also to make sure that it keeps the user safe during machine operation.

An existing generic desktop application is also provided for programming, calibration, test and monitoring of the controller box. I created the variable and stored parameters for the system as specified in the requirements document. I reviewed the parameter settings and cross check with each and every components. I also had to make sure that there are proper parameter settings for all the components that needs it. There are also live data parameters which the application shows while the system is running. I had to make sure that the necessary parameters specified in the requirements are good enough for the user to properly inspect the system during operation. I started the development by creating the necessary parameter interface necessary for the application to extract the data from the controller box.

2.2 Project 2

The controller application mainly provides the variable rate for the system and actuates the motors that will control the seed flow of a planter machine. These motors can be driven by a signal with a varied duty cycle which will allow variable rotational speed. These motors are installed in a retrofit planter system that has gears which controls the flow of the seeds from a tank going to the tubes that essentially throw seeds on the ground.

In this more schedule-aggressive project, I utilize the same controller box but this time its a system design for a planter implement. My main part is still in controller application development. Input, output, and control modules are still present but there are now two or more controller boxes.

The project comes with a need to interface to a GPS receiver. An existing GPS library software was used but I needed to make it work for this project. The output lines are now controlling motors instead of valves. I first used open loop control and eventually used close loop control in the next phases of the project. The control module implements a lot of algorithms and calculations to make the machine far more efficient than the traditional planter equipment. I also created code level testing using mock applications. This helped me to verify my calculation implementations right at the code level.

This project also comes with the mobile application that connects to the controller. I worked closely with the counterpart application developer to create the product. I defined and implemented the mobile application and controller application communication requirements. It made the unit test level complicated. I created a test setup in which two controller unit is attached with a GPS receiver and the mobile application controlling the system. This test was emulated by placing the setup inside the car, making the GPS receive actual movements and thereby simulating the planting tests.

2.2.1 Analysis

The project has already a requirements document which defines the control unit components. It also details out the algorithm and calculations that the controller has to do for the variable rate seeding control. At that time, before the project can proceed, there are a few more things that has to be filled in. There must be a way to calibrate the seeding rate of the motors and conduct experiments to validate whether the pulse-width modulation (PWM) would provide a suitable motor speed range to suit the requirements for variable seed rate control. I went through the design and synthesis, implemented the software controller code, and conducted test and experiments.

2.2.2 Design and Synthesis

To kick start the motor speed investigation, I implemented a basic controller application that drives one signal output line going to the motor. In theory, DC motor speed control can be done by armature control which either varies the armature voltage or the external armature resistance. Programming wise, the controller unit core drivers has been given better options for armature voltage control. I used two types of implementation. The 1st implementation drives the output signal of the controller with varying on and off times. This resulted in a not so smooth speed control due to the code relying heavily on delay timing execution. The next implementation uses the built-in core driver duty cycle variation. PWM essentially varies the average voltage level used to drive the motors which is basically the armature voltage control. Also, the frequency output signal of the controller is kept fixed as the duty cycle is varied. The series of experiments and tests I conducted showed that with the use of PWM, the motor gives enough rotational speed range suitable for the planter control.

Motor calibration aims to find out the number of seeds that comes out of the planter in every second. An assumption has been made that the seed rate increases linearly as the motor speed rate is also increased. Again, I designed the calibration state machine module in the software which follows directly to the calibration procedure. I created the start and stop calibration procedure and, timers in the code to measure the time elapse and a automated calculation that will output the result in number of seeds per second. This number was then used for the variable rate calculation. Actual tests were conducted on-site the customer location.

2.2.3 Implementation

I wrote the controller application code. This code together with the core driver runs in the micro controller unit. The micro controller is the heart of the controller box. It's the brain of the planter system. The retrofit planter is the muscle where the motors are located. The software is the intelligence of the system that implements all the logic for control. Given the based controller code, I implemented all the algorithms and calculation needed by the system. I designed and implemented a state machine that ties up user inputs, sensor readings, error statuses, and variable rate controls. I then created state transition conditions for each of the state. The output condition for each state are also implemented. Part of the output module are the actual turn on and turn off of the motor using the calculated duty cycle.

2.2.4 Testing

In this project, I was required to do bench testing before deploying the controllers to the actual machine. The tests are conducted on a unit level and system level. To do the system level testing, the sensors are simulated using a Testjig which is a external unit that simulates the signals such as voltage output of sensors and frequency values of speed sensors. The output lines of the controllers are tied with light emitting diodes (LED) to visualize the output ON and OFF states. I used external tools such as CAN analyzers to understand and decode the signals coming out of the CAN modules, oscilloscopes to understand and test waveform signals in the sensors and multimeters to read voltage, current and frequency levels. Some of the unit level testing are executed with the help of this tools. When I need to simulate sensor level to validate a feature, I used the test jig. Other unit test are executed in the code level. I run the code on the debug mode and emulates the system using software mocks.

The prototype motors doesn’t have frequency feedback signals that’s why validation was done by manual shaft rotation count. But on the 2nd phase of the project, the motors now comes with frequency feedback signal encoders. Frequency feedback is read and rpm is calculated based on the pulses-per-revolution represented by the number of gears inside the motor.

2.3 Project 3

My third major project is a controller application for forage harvester. Like the 2nd project, the system requires more input and output devices requiring the need for two controller units, a slave and a master controller. The slave controller unit is a smaller subset of the main controller unit. The two unit talks to one another using J1939 protocol. The project makes use of the Virtual Terminal (VT) display on the tractor. The display device serves as the main user/operator interface to the controller units. So for this project, I was tasked to continue the controller application development as well as the Graphical User Interface (GUI) running on the VT.

At that time, a specifically-designed PID has to be developed to achieve the requirement. I was assigned to also implement complex algorithms such as metal and stone detection. I also had the chance to widen my knowledge on ISO 11783 standard by implementing auxiliary control feature, virtual terminal button interfaces, and virtual terminal audio control in our software libraries - a first among all other projects in the company.

2.3.1 Analysis

The motor speed control is a critical element in this project. The motor is tied to a proportional valve wherein the hydraulic fluid allows the slowing down or speeding up of the motor speed. I assessed, by developing a cascaded PID closed-loop control, the feasibility of maintaining a steady motor speed with the external disturbances in the form of varied temperature on hydraulic valves, effect of engine throttling, vibrations in the implement, and effect of temperature of the surroundings.

Another critical requirements of the system is to allow the customer to do fast pressure sensor data logging. I was tasked to come up with software design that would allow such capability given the limitations of the current controller driver and libraries.

2.3.2 Design and Synthesis

I was responsible for the rest of the software implementation up to the completion of the requirements. This includes the code development of master controller application, slave controller application, VT graphical-user-interface (GUI) software, and the configuration and development of the PC-based software tools.

2.3.3 Testing

I was tasked to design a test setup that would allow me to test the cascaded PID, the metal and stone detection algorithm, and the fast data logging. Using the provided sample unit from the customer, I created a bench test setup that would allow me to test the cascaded PID. I was tasked to come up with a setup that would allow simulating the actual machine. I came up with setting up the master and slave controllers, a unit test jig configuration that simulates the sensor signals, and a set of tools such as multi-meter, oscilloscope and signal generators wired and connected for bench simulations.

2.3.4 Implementation

Like the previous projects, I divided the entire controller application into modules such as input, output, control and communication. These modules are further subdivided into sub modules that corresponds to a certain function that implements the requirements. I also added extra helper functions that would allow monitoring and testing the functionality of the application. I also implemented the complex algorithms such as metal and stone detection and cascaded PID for feedroll motor control.

I developed the separate VT GUI software which contains the graphic display for the machine operator. This display software still ties up to the controller unit which basically contains the logic implementation. I designed this VT module on the controller to comply with the ISO 11783 standard.

3 Practical Experience

3.1 Visits to Existing Engineering Works

Customer visits are vital to our software development. This is also an opportunity for me to learn the product further. I realized our software designs better as I saw how the actual machine operates. At the customer site, I learned real world test scenarios, I've learned the system design limits, I've realized additional required settings at the software level, conducted live debugging and make the system work for the 1st time. In exchange, I teach the customer how to operate the controller units, I teach the customer how to calibrate the machine, I teach them how to update the software on the controller units, and I also instruct them how to do basic diagnostics and sometimes advanced configurations on the machine.

I’ve also visited the customer's manufacturing site where the machines are built and assembled. At the site, I've done the Proportional-Derivative-Integral (PID) Controller tuning, motor setup and calibration, and functional testing for each and every testable component of the system at that stage. In those visits, I’ve learned so much on the basics of PID Controllers. I’ve realized how each of the P, I, & D component affects the feedback of the system. I've also come to see and visualize the motor control for the planters. I've seen the bins being loaded with actual grains and basically real-world machines being tested for the first time.

3.2 Application of Equipment as Part of the Larger System

The controller application I designed is just a part of a huge agricultural machinery. The control application software has to work as per the specified requirements so the machinery would be able to fulfill what it is intended for. This is also the part where I realized that a simple mistake in voltage reading in one of the inputs can largely affect the intended control. But through careful simulations, unit and system level testing on the bench and on the actual machine made my software implementation working as per design.

In all of the projects, I needed to understand how the entire system should work. One example is when connecting the controller unit from the implement to the tractor via ISOBUS. For project 2 and 3, I had to understand a lot of things such as the standard for ISOBUS connector, the proper CAN termination, the standard commands and responses to communicate to the VT device on the tractor, and extract GPS information provided by the tractor ECU.

3.3 Opportunities to Observe the Limitations of Practical Engineering and Related Human Systems

One of the interesting limitations I observed was about the hysteresis of hydraulic valves. On this project, the auger is driven by a proportional valve. This part can move in two opposite directions, fold and unfold. Applying constant duty cycle on driving a proportional valve doesn't always stop the auger at the same point and the same manner every time. It is almost always difficult to find the optimal duty cycle that will provide the most acceptable auger movement. This is where open loop controls using intelligent algorithms such as PID becomes helpful.

In project 3, using speed parameter in speed PID control looks straightforward at first. When we run the actual machine, the noise signal is way too far from a smooth sinusoidal signal. My basic PID implementation wasn't sufficient at all. I had to design ways to average the signal just enough but not too long so I could catch speed changes for my PID to react fast enough.

In the same project 3, there was detection algorithm design that uses averaging of signal and using standard deviation to catch sudden change in voltage levels. We thought that the 1st design is sufficient. But with further testing by the customer, it was found out that the noise signal coming from the detection device is amplified by faster motor speeds. So, we had to revise the design and I had to change the code implementation to consider the motor speed.

3.4 Opportunities to experience the significance of time in the engineering process

Being the main software controller application developer of this project, I was always involved in schedule estimates. I have done the schedule of this project based on the implementation design blocks. Again, basing on the requirements document, I further divided those major blocks into smaller functional units. I basically identified what are the things that I have to get done for a specific function. It then gives me a better idea how many hours I'll take to implement those. The idea was to break down the entire implementation into modules where each module are divided further into sub modules. Each sub modules are further divided into functionalities. The same 'Work Breakdown Structure' approach is used to estimate the testing.

4 Engineering Management

I was exposed to various areas of engineering management as follows:

4.1 Planning

In project 2, at the second phase of implementation, I created a high level plan on allowing our controller application to communicate with the task controller unit inside the virtual terminal device. I laid out the basic things that has to be done and has to happen before we could satisfy the customer requirements.

In another project not specified above, I've co-created the basic solution and design for a simplified version of my project 3. I've also opened up possible issue that needs to be addressed in the design of the project.

4.2 Scheduling

In all of the three projects, I've been involved heavily in the schedule creation. On a high-level view, I defined the things that has to be done for each of the requirements and the testing effort that has to be considered. I then come up with initial schedule estimate where I finalized them later with the project manager's approval. I created a checklist of what needs to get done for the project completion including minor details of issues that needs to be address and indicated the hours estimate to complete each task.

4.3 Budgeting

The schedule I created for every project were taken into account for the resource hours costs. I'm also giving inputs to the project manager about the things that I will be needing to complete the development, the equipment for testing, and the additional software tools that is needed.

4.4 Supervision

Being one of the early members of the development team, I'm always involved in code reviews. In a simpler version of project 3, I closely guided another developer in completing that project as it reused much of the code implementation that I've done.

4.5 Project Control

In most of my project, being the controller application developer and secondary to the project manager, my inputs are being considered for major design issues. In project 3, when it was decided to use a different way of data logging, I was trusted by the project manager to take care of implementing the best possible solution. In another case, a redesign of the virtual terminal was done. I discussed the best possible approach to come up with the control solution. I also talked directly to customer and decide when would be the dates for the software drop and when we will be able to deliver the fixes and new features.

4.6 Risk Assessment

Risk assessment are oftentimes triggered in the middle of the implementation phase. In the three projects I've held, I always give a heads up to the project manager of the risks that we're about to face. Whether it be a design or schedule issues, I'm giving my initial assessment of the problem and then propose solutions that we can use to solve the problem.

In project 1, while in the middle of implementing the operator controls, I recognized the need to have the physical joystick device before the 1st launch to customer. At that time, I was informed that we won't be receiving the device at the intended schedule. So, I speak to our manager and opened up the problem. I told them that the joystick would be critical for operation and safety. I suggested if there was a way to create a software that would simulate the joystick behavior. The team pursued the suggestion and was able to come up with that software and we were able to continue the development without the actual joystick device.

In another case, in project 2, I was in the middle of completing the project 3 while the next phase of project 2 is on its way. I immediately notified my project manager of the risk in the schedule. That time, project 1 is idle but still I informed my manager that a sudden request for next phase of project 1 would put risk in the completion of project 2 and 3. I told him that it would be better if someone would start taking over the project 1 so that I can handle the other two project's deadlines. I decided to give up project 1 since it was an easier project to take and that the embedded side is pretty much completed.

5 Communication Skills

Via teleconferences, I walk the customer through each of the basic functions of the unit. I also do further collaboration to the customer and discussed about the possible future enhancements and sometimes also talk about design implementation specific in any case some of the first function designs didn't work very well. For most of the projects I've held, early customer engagement happens as software is being developed. It includes confirming the design while it is being implemented and discussing possible implementation issues along the way. A lot of discussions happen via phone calls and email exchanges.

There are also times that I speak to customer while we are doing the debugging via phone. I will instruct the customer while they execute it and see the results. For a few customers, via desktop sharing, there were times I've been asked to look at their code and help them solve a certain problem.

6. Professional and Ethical Responsibilities

6.1. The responsibility of the engineer to the public.

I designed the controller application with error handling conditions such as circuit protection which detects short circuit, open circuits, and overload protection. All of the three projects I created follows a state machine design approach wherein proper states are defined to handle system failure conditions.

In all the controller application, I've added proper configuration for the PC-based software tool that would allow the user to properly diagnose the system on their own. I also implemented application side debug and data logging that helps capture data while the unit is in the field.

In every software release, a detailed software note is provided that instruct the user on how to set the system parameters and basically operate the machine.

6.2. The responsibility of the engineer to the profession.

In those parts of the software code that follows a certain standard such as ISO 11783-6, I reviewed the standard carefully and repeatedly to make sure that I understand how that certain part is to be designed. I do not simply copy an existing sample code developed by other members but also try and understand the underlying concept and do my own analysis.

In the office, I've been an active advocate of professional registration to my fellow immigrants in the production team. I've even offered to support them in the academic assessment path. Being a passer of Fundamentals of Engineering, I've encouraged some of them to follow the same path I took.

6.3 The responsibility of the engineer to the client and/or employer.

In doing technical support to customers, I always seek the approval of my supervisor or project manager for any details that I will be giving to the client. Likewise, I ensure that the requirements and design documents from clients are well secured.

Supervisor Agrees: Maurice has been an excellent member of our engineering team at JCA Electronics. He has been exposed to all areas of the engineering process in his role as a software developer at JCA and continues to develop as a responsible and capable professional.

Mentor Agrees: The JCA engineering team is relatively new and for this reason it regularly requires engineers to work on problems that may extend beyond their main core expertise. Maurice has shown a great ability to take on challenges that go beyond his main responsibilities as a software engineer in training. He is always willing to learn from other experienced engineers and is always available to help and mentor new team members.

---

2. Please check the following options that apply:

2.1: During this reporting period, I have applied theory in:

- ✓ Analysis/Interpretation
- ✓ Project Design/Synthesis
- ✓ Testing/Verification
- ✓ Implementation

Supervisor Agrees.
Mentor Agrees: Maurice's contributions in the Engineering team span throughout the whole design cycle, from the preliminary phases when proposals are being put together to supporting projects that are already in production.

2.2: I have obtained experience by:

- ✓ Studying or being exposed to existing Engineering works
- ✓ Applying Designs as part of larger systems
- ✓ Experiencing the limitations of Engineering designs
- ✓ Experiencing time as a factor in the Engineering process

Supervisor Agrees.
Mentor Agrees: Maurice has participated in projects that have exposed him to numerous challenges both technical and non-technical.

2.3: I was exposed to the following areas of Engineering management:

- ✓ Planning
- ✓ Scheduling
- Budgeting
- ✓ Supervision
- ✓ Project Management
- ✓ Risk Assessment

Supervisor Agrees.
Mentor Agrees: Maurice works very closely with project managers, and application/systems engineers that oversee these tasks.

2.4: I was required to make decisions based on professional and ethical responsibilities to:

- ✓ The Public
- ✓ The Profession
- ✓ The Client and/or Employer
- ✓ Co-Workers
- ✓ The Environment

Supervisor Agrees.

Mentor Agrees: Maurice has always conducted himself in a responsible and ethical manner. He is not afraid to identify risks and raise concerns during the development of the projects. This is very important for JCA, especially with software engineers as they are the ones directly shaping the behavior of the machines be build which will be ultimately used by the public.

3. Describe any activities that have improved your Communication, Teamwork, or Interpersonal Skills in the following areas:

Oral Presentations:
Under the forage harvester project of JCA, I conducted a presentation within the entire engineering team after doing a customer visit. I presented the data we gathered from our first run of the software on the actual unit. I detailed down my test results by presenting graphs of the data and the changes we've done in the implementation of one of the algorithms.

Written Documents:
As the main developer for the 3 projects mentioned, I created customer manuals such as controller re-programming, calibration procedures, basic controller operations and bug fix analysis. More often, I create documents which contains specific instructions on how to operate and configure the software that I've created for the project. I've done a few revisions of the requirements documents as well. Also, I've been a regular contributor to our engineering development wiki. This pages are being used by the entire engineering team.

Interaction with Others:
Almost every week during this time period, I talk to customers to update them of the software development status. I speak daily with my workmates to discuss technical things including design discussions, debugging, solicit advice to solve project-specific problems, requests for some technical information and even teach new members on the team on different technical matters.

Other:

Supervisor Agrees.
Mentor Agrees: I have used and consulted some of the documents that Maurice has put together and I have also attended some of his technical presentations. During these opportunities he has always shown good communication skills.

---

4. During this period, I had to consider the social implications of my work in the following areas:

One big benefit of working in a company which applies technology in agriculture is that the products we make has enormous impact to mankind and society. The farm equipment that we build together with our customers helps farmers and food producers to lower down their cost, do their farming with more efficient and less labor and essentially produce more yield. The above mentioned projects namely Grain Cart Controller Application and Variable Rate Seeding system are two best examples of this. These equipment makes farming process easier. In turn, it helps lower the cost of food for consumers. The forage harvester is one example for processing livestock foods better which in turn produce quality meats. Better farming technology helps sustain adequate food for mankind.

Supervisor Agrees.

Mentor Agrees: We are all very proud of the work we do at JCA as we understand the benefits that the applications we develop bring to society. Maurice has had an opportunity to work on projects that bring significant efficiencies to the agricultural market.

---

5. Examples of my ability to work effectively as part of a team, during this period, include:

With my years of experience and passion in embedded software development, I was able to learn and grasp the technology that we use easily. It allowed me to handle more projects within a short period of time. Apart from the projects mentioned above, I'm also dealing with other minor projects and help sustain old projects. The ability to multi-task is a big deal in our engineering team. Being one of the first few developers in the team, I'm also able to share my knowledge and help train newer developers. In fact, some of my past written software codes are already being used to build new projects.

Supervisor Agrees.

Mentor Agrees: Maurice always makes himself available to train and help new hires.
He has also demonstrated a great ability to work within a multidisciplinary team such as the JCA engineering team.

---

6. Examples of my ability to assume responsibility include:

In project 3 stated above, when the project was turned over to me, I assumed full responsibility of the controller application development. I reviewed all the existing requirements and design documents and made my way into completing the project. I also didn't hesitate to learn the new software that is needed to continue a part of the system software. I didn't even rely on asking the previous engineer involved but tried to learn the existing software code as fast as I could.
In one other product of the company where it was based on one of my current project, I let myself involved in the design and code reviews of the product. I suggested ways on how to address the design. I also guided and taught the software developer handling the project.

Supervisor Agrees.

Mentor Agrees: Developers at JCA are often required to work unsupervised during long periods of time, amusing full responsibility of the project during these stints.

---

7. I have shown progress since the last report (where applicable) as follows:

From my previous work, I had widen my technical expertise by having the experience to develop embedded control applications for agriculture. Before, I was part of a team developing platforms for different applications and now I'm in the side where I used the platform developed by the company to create a more direct application. I also gained new knowledge by developing software which uses J1939 and ISOBUS standards. Also, I learned how to use PID on a real-world application.

Within the short time period of dealing with control systems, I've been able to do a complex project that implements special controls - project 3. To date, the VT display control of that project has is the most complex among other projects in the company. I've made some VT control implementation where no other projects have ever used.

I've dealt with multiple projects at a time. There were instances that I have to switch from one project to another within two to three days multiple times.

Supervisor Agrees.

Mentor Agrees: Maurice's experience prior to JCA was focused mainly in the development of "low level" software. In his current role, he has been able to expand his skillset to "high level" tasks and other engineering disciplines.

---

8. I feel myself to be lacking in exposure to, or requiring improvement in, the following areas:

I'm confident that as I continue my work as a software developer, I'll be learning more and more on controller systems application and basically add more years in my technical experience. Right now, I'm dealing more on the embedded side. I believe more exposure to application development such as android development would widen my technical capabilities. Owning and designing a project from requirements generation to execution or a minor role in the project planning and design would benefit for my professional development.

Supervisor Agrees.

Mentor Agrees: The JCA engineering team is growing rapidly and we will rely heavily in the experienced engineers such as Maurice to take on new and diverse challenges.

---

9. I would like to provide the following additional, relevant information:

Before joining this company, I don't really have much background in control systems as I've been focused on platform software and board support packages. With the opportunity given by the company, I've dive into learning basic and advanced control system designs. Starting from project 1 with simple controls, the project moved to next phases adding more sophisticated controls in the application. From project 1, I've been given the project 2 and 3 which implements more complex controls. These projects are also more schedule-aggressive. There are times that I've been challenged to handle project deliveries alternately. And in the recent months, I've been assigned with minor projects on the side while still implementing those major projects. And slowly I've been given a chance to guide new developers in the team to help them develop and debug control application software. Furthermore, some of the codes that I've written have been used to develop other projects.

Supervisor: Darcy Cook P.Eng. (#25060) (First Registered: Jan 1, 2011)

I declare that I have adequate knowledge of the engineering or geoscience work experience described in this report, and that I have met with this person regularly to review their experience and progress. I therefore make the following evaluation and recommendation regarding the reported experience:

Maurice had joined the engineering team at JCA when we had only about 5 or 6 people in the group, and now have grown to about 25. This change has required not only him to come up to speed in a short period of time, but also for him to provide guidance to new people joining the team. This has been a trial by fire for him, and he has met the challenge. I have seen real growth in the time that he has been with JCA. He had strong coding skills to start from his previous experience, but it was limited in terms of following through the whole engineering process from concept, planning, design, verification/validation, to commissioning. At JCA he has ben exposed to a wider set of engineering problems, heavy on control systems, and getting the experience to see the machines that is software is controlling. This moves the development from the abstract to the more physical, and can drive a closer appreciation of the safety implications, as well as benefits to society and the environment. Maurice is on his way to becoming a very capable and responsible professional engineer.

In my opinion, during this reporting period, (Jan 4, 2016 - Jul 31, 2017) (19 months), Maurice has completed an equivalent of *19* months full time engineering work experience.

Please show my comments to the applicant

Mentor: Jorge Viramontes Perez P.Eng. (#35283) (First Registered: Jan 30, 2013)

I declare that I have adequate knowledge of the engineering or geoscience work experience described in this report, and that I have met with this person regularly to review their experience and progress. I therefore make the following evaluation and recommendation regarding the reported experience:

During his time at JCA, Maurice has demonstrated a great technical depth that allowed him to lead the development and implementation of important projects.
Maurice's main expertise when joining JCA was in the development of lower level functions for embedded systems but with the support of senior developers he has rapidly developed a good understanding of industry standards (like J1939, ISOBUS, etc) and control concepts such as PID controllers.
In a very short period of time Maurice has evolved into an expert in certain technical areas and software development processed and he has become a mentor for new members of the team.
In addition to his technical capabilities, Maurice has demonstrated great project management skills as he's often responsible for the planning and management of several project phases and he is the primary point of contact for some of our customers.
Maurice has demonstrated good judgement as well, he's not afraid to raise his hand when he sees any risks or has any concerns regarding both projects and company processes.
I would recommend that Maurice receives full credit for the time period covered in this report.

In my opinion, during this reporting period, (Jan 4, 2016 - Jul 31, 2017) (19 months), Maurice has completed an equivalent of *19* months full time engineering work experience.

Please show my comments to the applicant
