**UNIVERSIDAD TECNOLÓGICA DE QUERÉTARO**

**CESEQ**



**Diplomado en Software Embebido**

Integrator Project – PID Control of a DC Motor

File Name: Software Development Plan/SWRA\_20191030.docx

DOCUMENT: Software Development

Document No. #CESEQ\_SDP\_001

Scrum master:

Ramírez Altamirano Irvin (Mabe)

Developers:

Nepomuceno Herrera Juan Luis (Mabe)

Ramírez Altamirano Irvin (Mabe)

Tapia Guzmán Francisco Javier (UTEQ)

Date (YYYYMMDD): 20190817

Version: 1.0.0.

Project Version: 1.6.0.

# Log

Document Version

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date (yyyymmdd) | Description | Reviewer |
| 1.0.0. | 20190405 | First release | Pérez, Adbeel |
|  |  |  |  |

Project Document Version

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date (yyyymmdd) | Description | Reviewer |
| 1.1.0. | 20190823 | Progress | Juan, Francisco, Irvin |
| 1.2.0. | 20190824 | Defined deliverables, estimates, planning | Juan, Francisco, Irvin |
| 1.3.0. | 20190830 | Design plan | Juan, Francisco, Irvin |
| 1.4.0. | 20190831 | Defined testing and releases | Juan, Francisco, Irvin |
| 1.5.0. | 20191004 | Complete review | Juan, Francisco, Irvin |
| 1.6.0. | 20191030 | Corrections | Juan, Francisco, Irvin |

# Index

Table of Contents

[1. Log 2](#_Toc23451293)

[2. Index 3](#_Toc23451294)

[3. Project Scope 4](#_Toc23451295)

[4. Deliverables 5](#_Toc23451296)

[5. Development methodology 5](#_Toc23451297)

[6. Estimates 6](#_Toc23451298)

[7. Planning 6](#_Toc23451299)

[8. Solving Problem Strategy 9](#_Toc23451300)

[9. Design 10](#_Toc23451301)

[9.1. Standards 10](#_Toc23451302)

[9.2. Naming conventions 10](#_Toc23451303)

[10. Testing 10](#_Toc23451304)

[10.1. Verification strategy (black box test) 10](#_Toc23451305)

[10.2. White box strategy 11](#_Toc23451306)

[11. Release 11](#_Toc23451307)

[11.1. Software Development Folder 11](#_Toc23451308)

[11.2. Throughput and Flash and RAM measurement 11](#_Toc23451309)

[12. Results 12](#_Toc23451310)

[13. Lessons Learned 12](#_Toc23451311)

# Project Scope

For the project the main objective is to control the speed of a direct current motor through the application of a square signal that will variate in his pulse width and the work frequency must be constant.

The work frequency must be in a range of 100 Hz to 1KHz. Once selected the work frequency, it must be established, with variants only in the “duty cycle”.

When using a Hall effect sensor coupled to the motor rotor, the motor speed must be measured, which will provide a series of pulses each time a complete turn is completed. Therefore, the higher the motor speed, the greater the number of pulses read and the lower the speed, the lower the number of pulses.

The power voltage of the power card must be 12 Volts.

The motor must follow the reference value, which will be given by an input of the control card.

The LCD or graphical interface must show the speed of the engine and SetPoint (both in RPM’s); as well as the percentage of work of the square signal.

*PROJECT STRUCTURE\ 1) Requirements\* *3. SWRA\_20190831.xlsx*

Functional requirements:

* ***Inputs:***
  + The output of the sensor should be a pulse train of a square signal with variable frequency and with a maximum voltage of 13.6 V. To measure the speed, it should consider the number of registered pulses, in the hall sensor, that happen in a period of 100 ms, average it with the one of the next 100 ms and, the result, will be the speed showed in the display.
  + The adjustment of the set point shall be done through a potentiometer.
* ***Outputs:***
  + The frequency of the PWM signal shall be 1kHz.
  + The power circuit must be powered all the time at 12 volts
  + The circuit must provide a standardized 3.3V output from the hall effect sensor.
  + The supply voltage of the fan motor must be 12 Vdc.
  + The motor operating time must be determined based on the operating time of the BOUT1 output, between On / Off, at a constant period Δt.
  + When the ignition time is varied On, the motor speed must be varied. The change in the On time can range from 0%, 25% ... to 100%, depending on the set-point setting.
  + The percentage of operation must be indicated on the display screen
* ***Display appearance:*** 
  + It shall have a suitable contrast and brightness.
  + It shall be readable.
* ***Control:***
* Proportional - Integrative - Derivative should solve all the tasks of the system related to the speed control of a DC motor.
* ***Operative Systems:***
* The structure of the operative system shall be handled by Threads.

No functional requirements

* ***Display appearance***
  + Adequate contrast.
  + Good lighting.
  + Use of clear and legible typography.
  + Adequate refresh rate to avoid seeing “glitches” or transitions screens.
* **Flowcharts**
  + The project must include the flowcharts of the control algorithm and the functions used.

# Deliverables

* Folder of the project in C language (including c, h and hex files).
* Documents (Software Requirement Document, Estimates file, Planning file, Design file, verification file, Functional testing file, Gantt Diagram, FMEA).
* Hardware if apply (schematic files, general draft).

**Traceability of deliverables:**

The traceability of deliverables is going to be implemented through the GitHub platform. The tickets will be designed every sprint planning and released to GitHub using the ZenHub plugin.

With this tool we can know what changes were made in the code and who made these changes.

# Development methodology

SCRUM Methodology

* Scrum board: ZenHub.
* Length of the sprint: 1 week.
* Schedule of the Meetings: Friday (2:00 pm – 2:15 pm).
* Positions: Product Owner (Luis Urióstegui), Scrum master (Ramírez Altamirano Irvin), Quality assurance (Nepomuceno Herrera Juan Luis), Developers (Nepomuceno Herrera Juan Luis, Ramírez Altamirano Irvin and Tapia Guzmán Francisco Javier).
* Planning board: ZenHub (go to GitHub for more details).

In this project the following epics are considered:

**Epics:**

* ADC, PWM and interruption modules.
* Display module.
* PID module.
* System integration.

# Estimates

* Estimates **SHALL** contain all the Inputs, like:
  + Hw Facts:
    - Board availabilty
    - Plant availability
    - PC availability
    - Osciloscope
    - Signal generator
    - Multimeter
    - Power supply
  + Activities Facts
    - Human resources
      * Product owner should be the tutor
      * Master scrum and developer (better called as leader)
      * Number of Developers: 3
  + SW Facts
    - Operative system form Renesas works.
  + Hw Assumptions:
    - Hardware damaged.
    - Laboratory time availability.
  + Activtities Assumptions
    - Team time availability.
    - Hardware in good conditions.
  + SW Assumptions
    - Code in C programming language, SW IDE or Hw platform unkown.
    - SW Module unknown.

# Planning

**Rolles and Responsibilities:**

* Product Owner: Luis Humberto Uriostegui
* Shall be able to know about the product.
* Shall be able to attend the needs about the project.
* Shall be able to make know the requirements to Scrum Master.
* Shall be able to remanage the project if is necessary.
* Scrum master: Ramírez Altamirano Irvin.
* Shall be able to manage his team.
* Shall be able to have a good communication with his team.
* Shall be able to make know to his team the requirements by product owner.
* Shall be honest with himself and his team to target the correct times to develop the project.
* Quality assurance: Nepomuceno Herrera Juan Luis.
* Shall be able to know the goals of Project.
* Shall design test to assurance to correct performance of the product.
* Shall be able to assurance the quality of product.
* Shall be able to make know to the team about issues resulting of test.
* Developers: Nepomuceno Herrera Juan Luis, Ramírez Altamirano Irvin and Tapia Guzmán Francisco Javier.
* Shall develop the code for correct functionality of project.
* Shall be able to do the tickets in the time sprint.
* Shall be able to do know about his issues from project in the scrum daily.
* Shall be able to report his result of every sprint.

**Times Estimated for task in the project**

**Epics:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epic | Ticket | Developer | Definition of Done | FW |
| Power  Module | ADC Initialization | Ramírez Irvin | Configuration of ADC | 37 |
| PWM Initialization | Nepomuceno Juan | Add the PWM configuration to generate a signal with a frequency of 1 kHz | 37 |
| Power Module Integration | Nepomuceno Juan | Modify the PWM module to control the  duty cycle with a potentiometer through the ADC module | 38 |
| Interruption Initialization | Tapia Francisco | With a push button (integrated in the board) simulate a hall sensor and turn on a LED each 5 interruptions. | 38 |
| Refactor of Power Module | Ramírez Irvin | Refactor of PWM and ADC submodule. | 40 |
| PWM Invertir Logic | Nepomuceno Juan | Add a formula to invert the logic of the percentage variable. | 40 |
| Input Capture Initialization | Tapia Francisco | Initialize the input capture module in period measurement mode. | 41 |
| PID  Control | PID Algorithm  implementation | Ramírez Irvin | Implemetation of a digital PID algorithm | 41 |
| ADC Data and RPM Average | Nepomuceno Juan | Take three samples of ADC data with a  sampling time of 100 ms and average the values. Take two samples of speed data with a  sampling time of 100 ms and average the values. | 43 |
| User  Interface | Display Implementation  with GUIX Studio | Ramírez Irvin | Display the next values on the screen:  duty cycle, measured speed and set point. | 43 |
| Modules  Integration | Threads Implementation | Tapia Francisco | Add a thread for ADC, Input Capture, PID and PWM. | 43 |
| System Integration  Without UI - Realese 1.0 | Ramírez Irvin | Integrate all the modules developed until now. | 43 |
| System Integration With  UI - Realese 1.1 | Ramírez Irvin | Integrate all the modules developed  until now. | 43 |

All task above are described in the tickets of github´s repository.

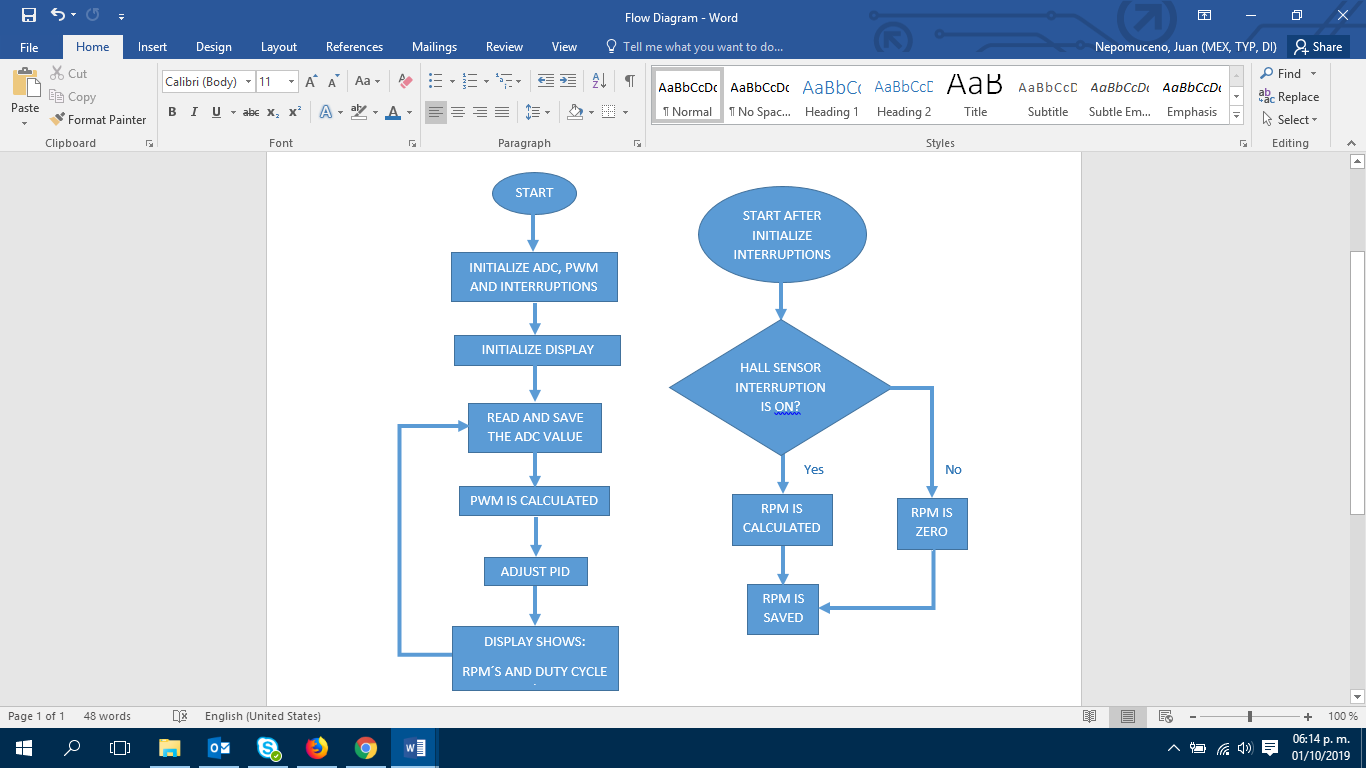
PROJECT STRUCTURE \2) Planning\7. Planning\_20190820.xlsx

# Solving Problem Strategy

PROJECT STRUCTURE \2) Planning\8. DFMEA\_20190905.xlsx

# Design

**Flow Diagram**



PROJECT STRUCTURE \3) Design\9. SoftwareDesignDocument\_20190405.docx

## Standards

In this case we use the C99 standard. For more details go to the document below.

Documentation\3) Design\ 9.1. SoftwareStandards\_20191026.docx

## Naming conventions

Documentation\3) Design\9.2. NamingConventions\_20191022.docx

# Testing

## Verification strategy (black box test)

Documentation\4) Verification\

## White box strategy

Documentation\4) Verification\

# Release

|  |  |  |
| --- | --- | --- |
| Date | HW Version | SW Version |
| September 20, 2019 | 1.0.0 | 1.1.0 |
| Ovtober 18, 2019 | 1.0.0 | 1.2.0 |
| October 24, 2019 | 1.0.0 | 1.3.0 |

The labels used in GitHub to denote a realease is simply “RELEASE”.

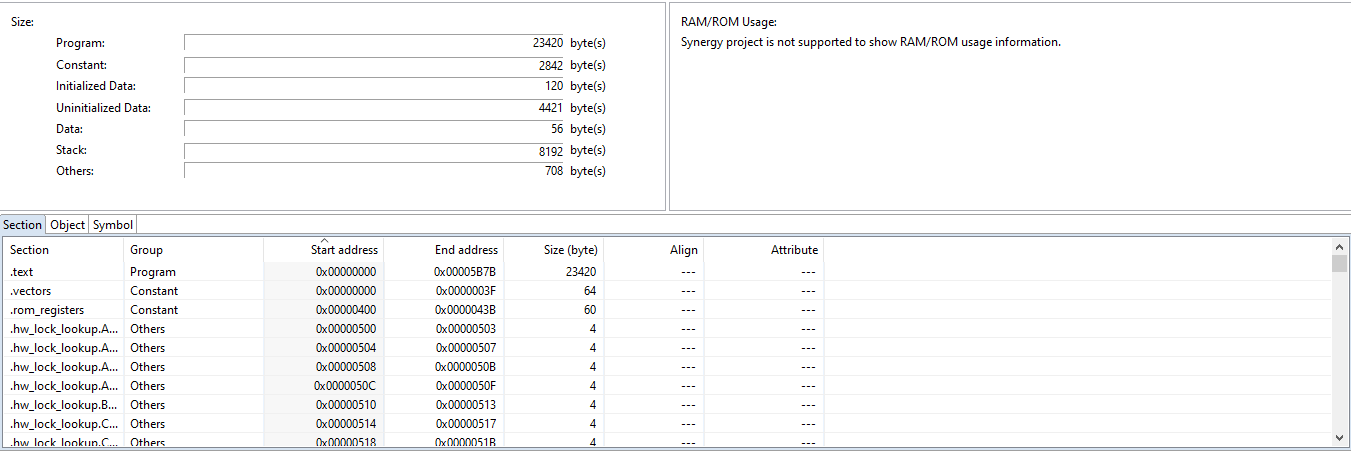
## Software Development Folder

<https://github.com/IrvinRamiA/Integrator-Project>

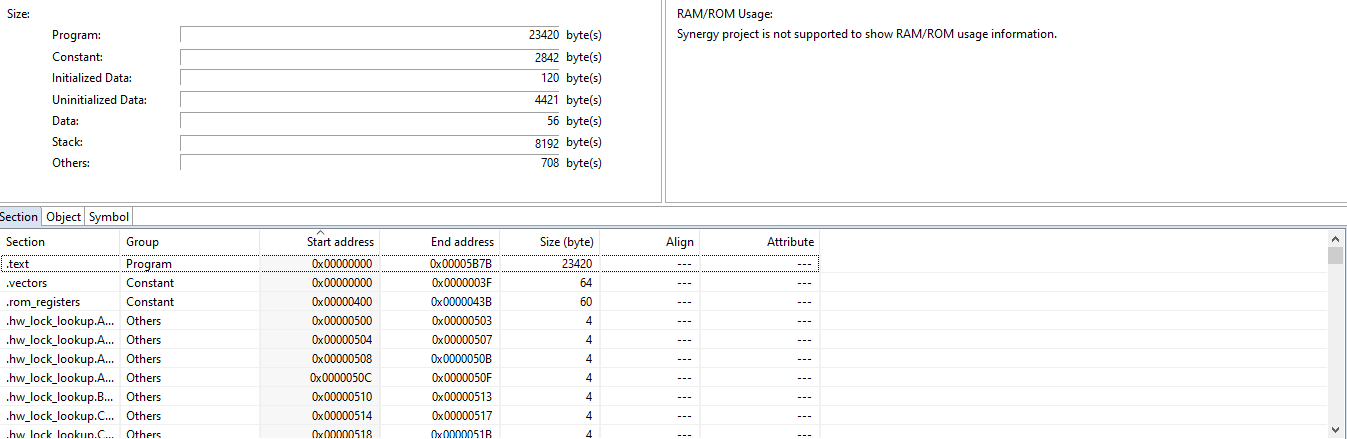
## Throughput and Flash and RAM measurement

PROJECT STRUCUTRE\4) Verification\ 11.3. ThroughputRAMFlash\_procedure

Power Module (V1.0.0)



System Integration without UI (V1.1.0)



Power Module (V1.1.0)

# Results

PROJECT STRUCUTRE\5) Results

# Lessons Learned

We believe that maybe the most important step in an embedded software project is the Planning, because it defines the development process and even what to do when a problem occurs. Although the dates and requirements can change, the planning is an useful design guide.