Software Specification

Virtual Meter for Semana I - <Gauge Name>

<Template Rev. 1.0>

|  |  |  |
| --- | --- | --- |
| Approval | Revision | Author |
|  |  |  |
|  |  |  |

# History of Changes

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | Details | Author | Date |
| 1.0 | Template created for the “Semana I” activity | E. Rios | 9/07/2015 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Purpose

The purpose of this document is to describe in detail the software requirements identified for the Virtual Meter project.

# Scope

This document describes the overall functional requirements of the Virtual Meter application that is loaded in the Raspberry Pi device as shown in the below deployment diagram.



The *Yazaki GUI* executable and its corresponding API is provided by the Yazaki Software Engineering group. The Virtual Meter Application is expected to implement the control logic stated in the product specification.

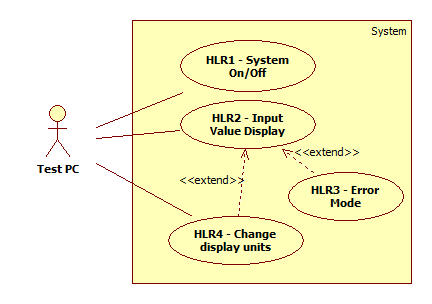
This document does not describe the requirements for the software products loaded in other devices such as the AVR CAN adapters or the PC test system.

Also, this document’s scope is the **<Meter ID>** gauge only. For details about other supported gauges, refer to the corresponding software specification documents.

**Caution: Please note that this template is an example of an “Oustide temperature gauge” that does not exist in the product specification. So adapt this document to match your gauge’s specification as required.**

# Functional Specification

The high level software requirements identified for this gauge are depicted in the below diagram with the HLR (High-Level-Requirement) prefix.



*Note: Although the above representation is a UML Use Case Diagram, the use of UML to describe the requirements is optional. Any other method to represent or describe the high level or low level requirements is also acceptable.*

The detailed requirements are specified below:

## HLR1 – System On/Off

### Power On

[LLR\_1.1] When the system is energized for the first time (power on), the IGN\_1 value is assumed to be IG OFF until the first CAN message is received.

### System On Sequence

[LLR\_1.2] When the IGN\_1 value changes from OFF to ON, the gauge is displayed with the pointer in the zero position.

### Normal Operation (IG ON)

[LLR\_1.3] While the IGN\_1 value remains ON, the target value is determined by the value of the OAT\_INPUT signal.

[LLR\_1.4] The current pointer position is determined as the “displayed value”, and the value of the CAN input signal is determined as the “target value”. When the target value is different from the display value, the target value is incremented (or decremented) by 1 lsb every screen refresh cycle until the display value is equal to the target value.

…

### System Off Sequence