



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

ΑΥΤΟΣΑΡ



Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ενσωματωμένα συστήματα αυτοκινήτου με βάση το πρότυπο Autosar Automotive Embedded systems based on Autosar Standard
Όνοματεπώνυμο Φοιτητή	Χρήστος Παπαδογεώργος
Πατρώνυμο	Κωνσταντίνος
Αριθμός Μητρώου	ΜΠΠΛ/12047
Επιβλέπων	Μ. Ψαράκης, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης

Νοέμβριος 2017

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Μ. Ψαράκης
Επίκουρος Καθηγητής

Χ. Δουληγέρης
Καθηγητής

Π. Κοτζανικολάου
Επίκουρος Καθηγητής

Περιεχόμενα:

I. Περίληψη/Abstract.....	4
1. Εισαγωγή.....	5
1.1. Εξέλιξη των ενσωματωμένων συστημάτων.....	5
2. Ενσωματωμένα συστήματα αυτοκινήτου AUTOSAR.....	6
2.1. Αρχιτεκτονική Autosar.....	8
2.2. Κύρια χαρακτηριστικά Autosar.....	9
2.3. Τεχνική επισκόπηση AUTOSAR.....	10
2.3.1. Ανάπτυξη Autosar Software components.....	11
2.3.2. Virtual Functional Bus.....	12
2.3.3. Αρχιτεκτονική του λογισμικού ECU.....	14
2.3.4. Η μεθοδολογία του Autosar.....	17
2.3.5. Δοκιμαστικοί έλεγχοι.....	18
2.3.6. Επιλογή Εργαλείων λειτουργίας και συντήρησης του Autosar.....	18
3. Μελέτη περίπτωσης.....	22
3.1. Περιγραφή.....	23
3.2. Αρχιτεκτονική του μοντέλου.....	24
3.3. Περιγραφή - Τεχνικά χαρακτηριστικά του μοντέλου.....	25
4. Συμπεράσματα - Περίληψη.....	30
5. Βιβλιογραφία	32
6. Παράρτημα.....	34

Ι. Περίληψη

Στη σημερινή εποχή η χρήση των οχημάτων είναι τεράστια ξεκινώντας από προσωπική χρήση μέχρι την βαριά βιομηχανία και τα μέσα μαζικής μεταφοράς. Συνεπώς η εξέλιξη που απαιτείται για να ικανοποιηθούν οι ανάγκες της σύγχρονης καθημερινότητας μας είναι μεγάλη. Στην παρούσα πτυχιακή εργασία μελετάται και αναλύεται ένα πρότυπο αρχιτεκτονικής ανάπτυξης εφαρμογών ενσωματωμένων συστημάτων το οποίο ονομάζεται Autosar. Το πρότυπο δημιουργήθηκε αποκλειστικά για ενσωματωμένα συστήματα οχημάτων. Αρχικά περιγράφεται η εξέλιξη των προγενέστερων συστημάτων του Autosar έτσι ώστε να γίνει κατανοητό από πού γεννήθηκε η ιδέα του Autosar και φυσικά τι σχέση έχει αυτό με τους προγόνους τους. Στη συνέχεια παρουσιάστηκε και αναλύθηκε το πρότυπο σε βιβλιογραφικό επίπεδο. Σε αυτό το επίπεδο περιλαμβάνονται η πλήρη αρχιτεκτονική σχεδίαση του λογισμικού από το επίπεδο του υλικού έως το ανώτερο επίπεδο λογισμικού, τα τεχνικά χαρακτηριστικά που υπαγορεύουν και προαπαιτούν τα όρια και τα στοιχεία για την ανάπτυξη των εφαρμογών, του στόχους και τους περιορισμούς. Επιπλέον μελετήθηκε η μεθοδολογία και τα εργαλεία ανάπτυξης και εφαρμογής από το στάδιο του ενσωματωμένου κώδικα έως και την προσομοίωση. Σε τελικό επίπεδο μελετήθηκε η πρακτική εφαρμογή του προτύπου, το οποίο πραγματοποιήθηκε μελετώντας το μοντέλο 'AUTOSARDemoMdl' της βιβλιοθήκης της Mathworks. Το συγκεκριμένο μοντέλο είναι αναπτυγμένο με το toolbox του MATLAB, το οποίο και περιέχει τα δυο τρίτα των εργαλείων που απαιτούνται για την ανάπτυξη του προτύπου. Παρουσιάζεται, αναλύεται και σχολιάζεται σε πρακτικό επίπεδο η ανάπτυξη του μοντέλου, καθώς και η εφαρμογή του σε σχέση με την αρχιτεκτονική του μοντέλου, τα τεχνικά χαρακτηριστικά που χρησιμοποιούνται σύμφωνα πάντα με όσα μελετήθηκαν σε θεωρητικό επίπεδο. Στο τέλος παρουσιάζονται προσωπικά συμπεράσματα παρατηρήσεις και σχόλια.

Abstract

Nowadays, using vehicles is very common and necessary to ease peoples' life, starting from personal use to heavy industry and public transport. Therefore, the desired levels of evolution in order to meet our everyday life needs are significantly high. In this project, the embedded systems application development architecture, called Autosar, is being analyzed and evaluated. The model is created exclusively for embedded vehicle systems. Initially, the evolution of the first Autosar systems is introduced to show how Autosar was originally developed and thereafter to be compared to its predecessors. Besides that, in this project I continue on giving a detailed theoretical specification for the model. That specification includes the complete architectural design of the software from the hardware level to the highest level of software, the technical features which determine the limits and the elements for the development of applications, objectives and any constraints. Further, a good description of the methodology and tools for development and implementation is given from the embedded programming stage to the simulation phase. In addition to this, I study in practice the application of the template which was done by taking the Mathworks 'AUTOSARDemoMdl' model (as found in the Mathworks library). This certain model contains two-thirds of the tools needed to develop and complete the template. Finally, I make my observations and conclusions in regards to the development of the model, as well as the application of the model in relation to its architecture and the technical characteristics used in theory to test or verify it.

1. Εισαγωγή

Στην εργασία που ακολουθεί αρχικά μελετάται και αναλύεται ένα πρότυπο αρχιτεκτονικής ανάπτυξης εφαρμογών ενσωματωμένων συστημάτων το οποίο ονομάζεται Autosar και αφορά αποκλειστικά, ενσωματωμένα συστήματα οχημάτων. Η παρακάτω εργασία χωρίζεται σε τρία μέρη. Το πρώτο μέρος της εργασίας είναι εισαγωγικό και περιγράφει την εξέλιξη των προγενέστερων συστημάτων του Autosar καθώς και μια γενική αναφορά σε αυτό.

Το δεύτερο μέρος της εργασίας είναι βιβλιογραφικό και εμπεριέχει μια πλήρη παρουσίαση και ανάλυση του προτύπου που μελετάται. Αρχικά παρουσιάζονται τα κύρια χαρακτηριστικά του προτύπου και στη συνέχεια παρουσιάζονται και αναλύονται τα τεχνικά χαρακτηριστικά ενός λογισμικού που αναπτύσσεται σύμφωνα με το πρότυπο.

Το τρίτο και τελευταίο μέρος πραγματεύεται την πρακτική εφαρμογή του προτύπου σύμφωνα με μοντέλο της βιβλιοθήκης της Mathworks το οποίο είναι αναπτυγμένο με το toolbox του MATLAB και επίσης περιέχει τα δυο τρίτα των εργαλείων που απαιτούνται για την ανάπτυξη του προτύπου. Παρουσιάζεται και σχολιάζεται σε πρακτικό επίπεδο η ανάπτυξη του μοντέλου, καθώς και μελετάται η εφαρμογή του σε σχέση με την αρχιτεκτονική του μοντέλου από όπου ξεκίνησε η δημιουργία του, σε θεωρητικό επίπεδο. Τέλος παρουσιάζονται τα δικά μας συμπεράσματα και σχόλια.

1.1 Εξέλιξη των ενσωματωμένων συστημάτων

Παρατηρώντας την εξέλιξη στις σύγχρονες μονάδες ελέγχου μηχανολογικών εφαρμογών γίνεται πολύ εύκολα αντιληπτό ότι όλες οι σύγχρονες μηχανολογικές εφαρμογές σε πάρα πολλούς τομείς της ζωής μας λειτουργούν με μικροϋπολογιστές από απλές οικιακές συσκευές, την ναυτιλία, την αεροναυπηγική και την αυτοκινητοβιομηχανία που είναι και το αντικείμενο που χρησιμοποιείται στη μελέτη μας. Πιο συγκεκριμένα ο μικροϋπολογιστής της εκάστοτε ηλεκτρικής συσκευής λειτουργεί όπως ο ανθρώπινος εγκέφαλος δηλαδή αντλεί πληροφορίες από τους αισθητήρες (ανθρώπινες αισθήσεις) και αναλόγως δίνει της κατάλληλες εκτελέσιμες εντολές.

Τα πρώτα ενσωματωμένα συστήματα αυτοκινήτου - Automotive embedded system (AES) εμφανίστηκαν στα αυτοκίνητα στις αρχές της δεκαετίας του '80 και είχαν αναλογική λειτουργία χρησιμοποιώντας μια ROM με προ εγκατεστημένες τιμές για την διαχείριση του κινητήρα και ονομαζόντουσαν EFI (Electronic Fuel Injection). Το κυριότερο μειονέκτημα των παραπάνω συστημάτων είναι ότι οι τιμές αφορούσαν έναν καινούργιο μηχανικό σύνολο που απέδιδε σε ιδανικές συνθήκες ενώ στην χρήση του οι συνθήκες διαφέρουν κατά πολύ και επιπλέον με την πάροδο του χρόνου υπήρχε φθορά των υλικών.

Η εξέλιξη των AES ήταν οι μικροϋπολογιστές που χρησιμοποιούνται ευρέως από τις αυτοκινητοβιομηχανίες από της αρχές της δεκαετίας του 90' έως και σήμερα. Ο μικροϋπολογιστής ενός αυτοκινήτου είναι βασικά κάτι αντίστοιχο με τον προσωπικό μας υπολογιστή χωρίς το λειτουργικό σύστημα του ηλεκτρονικού υπολογιστή που αλληλοεπιδρά με τον χρήστη. Οι μεγαλύτερες και πιο γνωστές εταιρίες ηλεκτρονικών συστημάτων είναι η Bosch, Denso, Magnetti Marelli κ.α. Τα AMS πλέον χρησιμοποιούνται για τον έλεγχο όλων των λειτουργιών του ηλεκτρικού συστήματος ενός οχήματος από τον έλεγχο λειτουργιών του κινητήρα ως και το πιο απλό πράγμα που είναι η ρύθμιση του καθρέπτη.

Το 1980¹ η General Motors παρουσίασε μια ECU η οποία βάσιζε για πρώτη φορά πολλές από τις λειτουργίες της σε μικροεπεξεργαστές με πρωταρχικό στόχο την ανάφλεξη. Ήταν η αρχή του τέλους για την μηχανική διαχείριση της λειτουργίας ενός κινητήρα οχήματος. Πλέον η ανάπτυξη των ηλεκτρονικών συστημάτων ενός οχήματος είναι μείζονος σημασίας ειδικά σε θέματα, αποδοτικότητας, ασφάλειας και κόστους.

¹ Kassakian, J.G; Wolf, H.-C.; Miller, J.M.; Hurton, C.J. (1996) Automotive electrical systems circa 2005 – IEEE Spectrum

Η απλή, σε σχέση με τα σημερινά δεδομένα, σχεδίαση, ανάπτυξη και κατασκευή των οχημάτων επέτρεπε τον ταυτόχρονο έλεγχο και εφαρμογή του λογισμικού απευθείας στο όχημα. Έχει παρατηρηθεί ότι αυτό είναι πλέον αδύνατο. Αρχικά προγραμματιζόντουσαν οι εργον στους μικροεπεξεργαστές, όμως η πολυπλοκότητα έχει αυξηθεί τόσο πολύ που κάποιοι αισθητήρες είναι σε τέτοιο επίπεδο όπου πρέπει να έχουν κάποιες λειτουργίες παρακολούθησης όπως είναι το ESP.

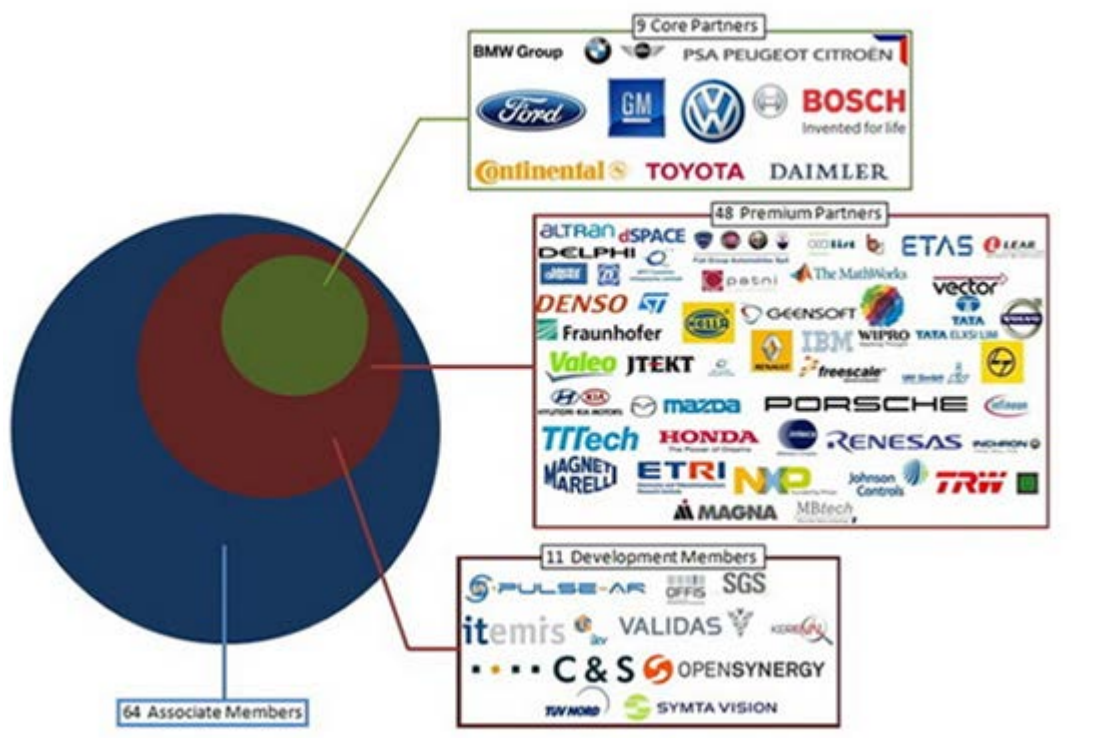
Η διαρκώς αυξανόμενη ποσότητα ενσωματωμένων συστημάτων, καθώς σήμερα τα οχήματα έχουν διαφορετική μονάδα ελέγχου για τον κινητήρα, κιβώτιο ταχυτήτων, συστήματα άνεσης, ασφάλειας και άλλα, γέννησε την ανάγκη για να γίνεται ποιοτικός έλεγχος στο λογισμικό πριν την εφαρμογή του. Είναι κοινώς αποδεκτό τα bugs σε ένα λογισμικό μπορούν να δημιουργήσουν πολλά προβλήματα στα υλικά που ελέγχει το λογισμικό, ακόμα και την αχρηστία του υλικού. Οι αυτοκινητοβιομηχανίες χρησιμοποίησαν την μέθοδο του HiL (Hardware in Loop), πρώτου το λογισμικό τοποθετηθεί στο τελικό προϊόν δοκιμάζεται σε εργαστηριακές συνθήκες σε πρωτότυπα, μέχρι να επιτευχθεί το επιθυμητό αποτέλεσμα.

Σε επίπεδο υλικού τα οχήματα πριν αποκτήσουν πολύπλοκα ενσωματωμένα συστήματα στον εξοπλισμό τους εφοδιάζονταν με απλά καλώδια για την απλή επικοινωνία των ηλεκτρικών εξαρτημάτων τους. Στη συνέχεια οι κατασκευαστές αναγκάστηκαν κυρίως λόγω κόστους, βάρους και πολυπλοκότητας να αναπτύξουν συστήματα διάυλων επικοινωνίας (Bus systems) με επικρατέστερο τελικά το CAN bus. Έτσι λοιπόν ο τρόπος επικοινωνίας είναι ενιαίος σε όλους τους κατασκευαστές οχημάτων διευκολύνοντας την κατασκευή τους, μειώνοντας το κόστος και αυξάνοντας την αποδοτικότητα.

Συνέπεια αυτής της εξέλιξης, της ανάγκης για άμεση επικοινωνία και συνεργασία όλων των υποσυστημάτων ενός οχήματος είναι η γέννηση του AUTOSAR, ένα πρωτόκολλο αρχιτεκτονικής λογισμικού για εφαρμογές σε Οχήματα στο οποίο συμμετέχουν οι κατασκευαστές αυτοκινήτων και οι προμηθευτές υλικών και εργαλείων. Κύριος σκοπός είναι η δημιουργία μιας υποδομής για την ανάπτυξη λογισμικού σε διάφορα επίπεδα για την αυτοκινητοβιομηχανία έτσι ώστε διαφορετικά συστήματα από διαφορετικούς προμηθευτές να συνεργάζονται, να είναι διαχειρίσιμα σε διαφορετικά οχήματα και κατά τη διάρκεια της ζωής τους να υπάρχει η δυνατότητα αναβάθμισης, αναπροσαρμογής και συντήρησης.

2. Ενσωματωμένα συστήματα αυτοκινήτου AUTOSAR

Το AUTOSAR (Automotive Open System Architecture) είναι ένα ανοικτό πρότυπο αρχιτεκτονικής λογισμικού για την ανάπτυξη ενσωματωμένων συστημάτων αυτοκινήτου. Δημιουργήθηκε επίσημα τον Ιούλιο του 2003 ενώ οι πρώτες συζητήσεις ξεκίνησαν τον Αύγουστο του 2002 από τους BMW, Bosch, Continental, Daimler Chrysler και Volkswagen. Πλέον συμμετέχουν όλοι οι μεγάλοι κατασκευαστές οχημάτων, ανταλλακτικών και εργαλείων.



Εικόνα 1 Μέλη του προτύπου Autosar²

Η τεχνολογία των υπολογιστών είναι πολύ ανεπτυγμένη συγκριτικά με την αυτοκινητοβιομηχανία. Η εφαρμογή της όμως στα οχήματα με την τεχνολογία που χρησιμοποιείται μέχρι και σήμερα δεν είναι εφικτή κυρίως λόγω του ότι η ανάπτυξη σε επίπεδο λογισμικού δεν ακολουθεί το ρυθμό των υλικών ή το λογισμικό στους υπολογιστές. Το Autosar γεννήθηκε από αυτήν την ανάγκη.

Οι κατασκευαστές οχημάτων έχουν συνεχώς απαιτήσεις ώστε να βελτιώνουν συνεχώς τα προϊόντα τους σε τομείς όπως η παθητική ασφάλεια, η περιβαλλοντική συνείδηση, άνεσή και η ψυχαγωγία. Πλέον εξελίσσονται και τα συστήματα ενεργητικής ασφαλείας με στόχο στο μέλλον να αντικαταστήσουν τον φυσικό οδηγό.

Είναι κοινώς αποδεκτό αυτή την εποχή από όλους τους κατασκευαστές που συμμετέχουν στον πυρήνα του Autosar ότι πρέπει να συνεργαστούν όλοι μαζί για να μπορέσει η βιομηχανία των οχημάτων να αναπτυχθεί. Έτσι λοιπόν έχουνε θέσει στόχους για να δημιουργηθεί ένα πρότυπο λογισμικού που θα ελέγχει τις λειτουργίες μιας ECU αυτοκινήτου σε πολλαπλά επίπεδα.

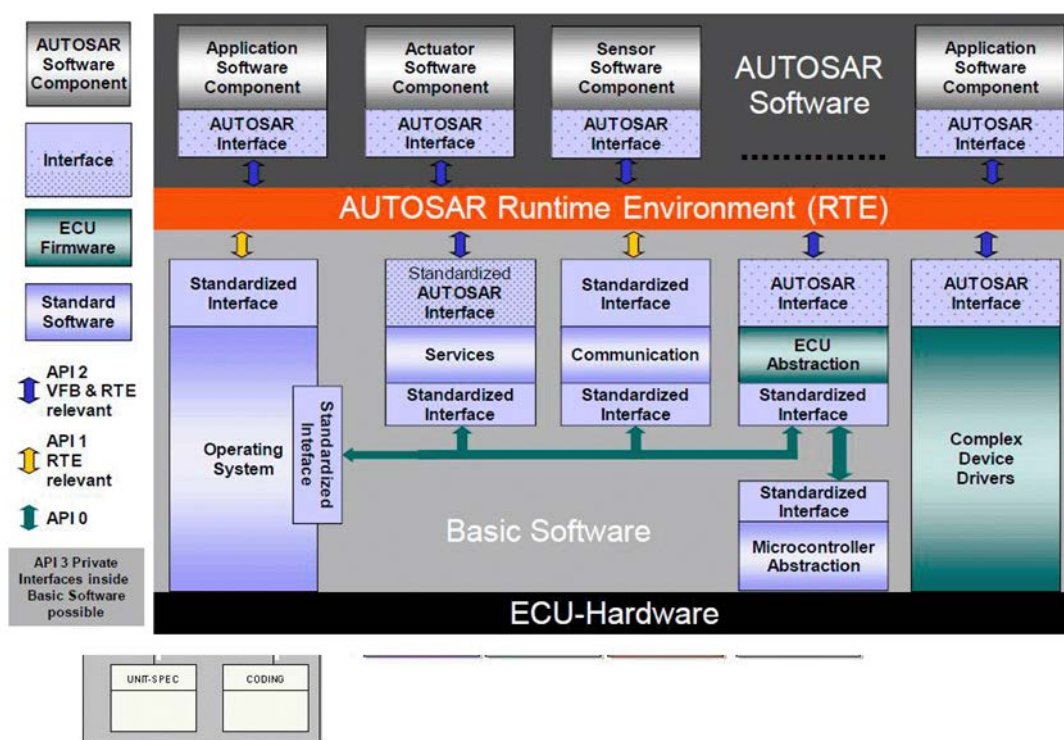
Οι κύριοι στόχοι που προέκυψαν από τη δημιουργία του Autosar αποτυπώνονται επιγραμματικά παρακάτω:

- Ορισμός και ανάλυση του ανοικτού προτύπου έτσι ώστε να είναι απόλυτα κατανοητό και κατ' επέκταση το λειτουργικό του επίπεδο να εκτοξευτεί.
- Δυνατότητα επεκτασιμότητας ενός λογισμικού σε πρότυπο Autosar σε οχήματα και πλατφόρμες διαφορετικών κατασκευαστών.
- Εύκολη μεταφορά και διαχείριση του λογισμικού.
- Η υποστήριξη διαφορετικών και λειτουργικών "κόμβων" είτε είναι προμηθευτής είτε πελάτης.
- Συνεργασία όλων όσων συμμετέχουν στον οργανισμό αλλά και όποιος θελήσει να συμμετάσχει μελλοντικά.

² <https://www.engineersgarage.com/articles/autosar-automotive-open-systems-architecture>

- Υποστήριξη προτύπων εφαρμογών οχημάτων και τεχνολογίες υψηλών προδιαγραφών.
- Το πρότυπο πρέπει να εξυπηρετεί ως μια πλατφόρμα όπου οι μελλοντικές εφαρμογές οχημάτων θα είναι ανεξάρτητες από τον κατασκευαστή του υλικού και θα ελαχιστοποιούν τα προβλήματα επικοινωνίας και συνεργασίας μεταξύ των δικτύων και των εφαρμογών.
- Ο κύριος σκοπός του AUTOSAR περιλαμβάνει όλη τη βιομηχανία των οχημάτων διότι χωρίς την συνεργασία και την κοινή αποδοχή ενός προτύπου η εξέλιξη, η συνεχής υποστήριξη και η ανταγωνιστικότητα των οχημάτων θα μείνει στάσιμη, και δεν θα μπορέσει να ακολουθήσει την εξέλιξη άλλων κλάδων. Παραδειγματικά παρατηρώντας την εξέλιξη των ηλεκτρονικών υπολογιστών τα προηγούμενα 10 χρόνια μπορεί να γίνει κατανοητό το πώς θα ήταν επιτυχημένη η πορεία του Autosar.

2.1 Αρχιτεκτονική Autosar



Εικόνα 2 Αρχιτεκτονική λογισμικού AUTOSAR.³

Αρχικά, παρατηρώντας την παραπάνω εικόνα τα βασικά επίπεδα της αρχιτεκτονικής του Autosar είναι χωρισμένα σε τμήματα, στο βασικό λογισμικό (Basic Software), και στο λογισμικό του Autosar (AUTOSAR Software) που περιλαμβάνει και το περιβάλλον λειτουργίας Runtime Environment (RTE).

Από κάτω προς τα πάνω στο πρώτο επίπεδο του βασικού λογισμικού της συναντάται τα επίπεδα Microcontroller Abstraction, Services Layer, Complex Drivers, ECU Abstraction. Microcontroller Abstraction: Στο συγκεκριμένο επίπεδο πραγματοποιείται η διαχείριση των μικροελεγκτών. Στο στάδιο αυτό το λογισμικό είναι εξαρτημένο από το υλικό αφού «κουμπώνει» σε αυτό, όχι όμως από την διεπαφή του υλικού αλλά με στόχο να ανεξαρτητοποιήσει όλα τα υπόλοιπα επίπεδα από την σχεδίαση του υλικού της ηλεκτρονικής μονάδας ελέγχου. Πιο συγκεκριμένα επικοινωνεί με το ενσωματωμένο λογισμικό που υπάρχει στη μνήμη των μικροελεγκτών για την παραμετροποίηση του σύμφωνα με το Autosar.

³ http://www.eetimes.com/document.asp?doc_id=1276593

Complex Drivers: Είναι το επίπεδο του λογισμικού που περιέχει όλα τα προγράμματα οδήγησης σημαντικών και πολύπλοκων λειτουργιών των μικροελεγκτών που λειτουργούν σε αισθητήρες και ενεργοποιητές. Για παράδειγμα μια τέτοια λειτουργία είναι ο συγχρονισμός της ανάφλεξης του κινητήρα, ο οποίος επικοινωνεί απευθείας με το υλικό, συνεπώς είναι άρρηκτα συνδεδεμένο με αυτό. Παρόλα αυτά ακολουθεί αυστηρά διεπαφές του Autosar.

ECU Abstraction: Είναι το επίπεδο του λογισμικού που φροντίζει για την ομαλή συνεργασία και σύνδεση μεταξύ του λογισμικού σε επίπεδο εφαρμογής με τους μικροελεγκτές και το υλικό της μονάδας ελέγχου (ECU). Η ανάπτυξη και η εφαρμογή είναι εξαρτημένη στο υλικό της μονάδας ελέγχου, περιλαμβάνει διεπαφές προγραμματισμού εφαρμογών (API) είτε με τους μικροελεγκτές της μονάδας είτε με προγράμματα οδήγησης περιφερειακών συστημάτων.

Services Layer: Είναι το υψηλότερο βασικό επίπεδο λογισμικού και λειτουργεί σαν ένα πλήρες λειτουργικό σύστημα που είναι μεν ανεξάρτητο από το υλικό και από το επίπεδο εφαρμογών αλλά κομμάτι αυτών δε. Περιλαμβάνει βασικές υπηρεσίες επικοινωνίας και διαχείρισης δικτύου του οχήματος, διαγνωστικές υπηρεσίες και μνήμης.

Στο επίπεδο του Autosar λογισμικού περιλαμβάνει διαχείριση των εξαρτημάτων λογισμικού που ενσωματώνονται και εκτελούνται στις εφαρμογές καθώς και περιβάλλον λειτουργίας (Runtime Environment) το οποίο παρέχει υπηρεσίες επικοινωνίας για την ανάπτυξη εξαρτημάτων λογισμικού σύμφωνα με το πρότυπο, τα οποία αφορούν ως επί το πλείστον τα εξαρτήματα λογισμικού που είναι ενεργοποιητές και αισθητήρες. Εφαρμόζεται μαζί με το λογισμικό που αναπτύσσεται αποκλειστικά για την μονάδα ελέγχου που προορίζεται. Επιπλέον περιλαμβάνει πρωτόκολλα επικοινωνίας με τις διεπαφές προγραμματισμού εφαρμογών (API) του βασικού λογισμικού.

2.2. Κυρία χαρακτηριστικά του Autosar

Ακολουθώντας απαριθμούνται και αναλύονται τα κύρια χαρακτηριστικά του προτύπου έτσι ώστε το σχέδιο να φτάσει στο στόχο του, που όπως προαναφέρθηκε είναι η καθιέρωση του ως ή βάση για ανάπτυξη συστημάτων αυτοματισμού στα οχήματα.

Δημιουργία σταθερών διεπαφών και πρότυπου.

Αρχικά πρέπει να οριστούν τα επίπεδα σχεδίασης του λογισμικού εξαιρετικά αναλυτικά με γνώμονα την πολυπλοκότητα που πρέπει να έχει το τελικό αποτέλεσμα. Οι δυνατότητες του τελικού χρήστη να έχουν τους ελάχιστους δυνατούς περιορισμούς. Αντίστοιχα οι απαιτήσεις του προτύπου σε υλικό πρέπει να είναι εξαιρετικά απλές, συγκεκριμένες και πολύ καλά δομημένες έτσι ώστε η δουλειά των προμηθευτών να απλοποιηθεί. Συνεπώς οι προμηθευτές που θα παράγουν προϊόντα θα είναι περισσότεροι δημιουργώντας ένα ανταγωνιστικό περιβάλλον σε τεχνικό και οικονομικό επίπεδο το οποίο θα ανθίσει την πρόοδο και την καινοτομία.

Ποιο συγκεκριμένα, σε τεχνικό επίπεδο, το Autosar έχει θεσπίσει κατεύθυνση να οριστούν τα περιεχόμενα του λογισμικού (SW components), διεπαφές προγραμματισμού εφαρμογών (API) και τύποι των δεδομένων (data types) με αδιαπραγμάτευτο στόχο την επεκτασιμότητα, την παραμετροποίηση και την συνεργασία μεταξύ διαφορετικών συστημάτων.

Χαρακτηριστικά Runtime environment.

Η δημιουργία του προτύπου RTE βασίζεται στα παραπάνω χαρακτηριστικά σχεδίασης του λογισμικού. Σε επέκταση, όλοι οι κόμβοι επικοινωνίας της πλατφόρμας πρέπει να επικοινωνούν μεταξύ τους αφογά καθώς η θέση του και το έργο του είναι να επικοινωνεί και να τρέχει τις διεργασίες μεταξύ των περιεχομένων του λογισμικού (SW components), των διεπαφών προγραμματισμού εφαρμογών (API) και των τύπων δεδομένων (data types). Επιπλέον, το RTE οφείλει να διευκολύνει τον τελικό χρήστη έτσι ώστε να ενσωματώσει το λογισμικό που έχει αναπτύξει σύμφωνα με το πρότυπο απλά, εύκολα και γρήγορα διότι σε διαφορετική περίπτωση η χρήση του προτύπου θα είναι δύσκολη με συνέπεια την απαξίωση του.

Δυνατότητα ελέγχου.

Με την έκρηξη της τεχνολογίας, τους γρήγορους ρυθμούς παραγωγής νέων προϊόντων και την αυξημένη ανταγωνιστικότητα η σημαντικότητα των ελέγχων προτού προχωρήσει η παραγωγή έχει γίνει επιτακτική έτσι ώστε η ποιότητα να παραμείνει υψηλή και το κόστος να περιορίζεται. Συνεπώς το πρότυπο δεν μπορεί να αγνοήσει αυτό το τόσο σημαντικό κομμάτι του

σχεδιασμού. Σκοπός λοιπόν είναι να δημιουργηθεί πρότυπο το οποίο θα εκτελεί ελέγχους με σεβασμό στα δεδομένα, περιεχόμενα του λογισμικού, του RTE, την συμβατότητα του δίαυλου επικοινωνίας και των εφαρμογών που λειτουργούν εμβόλιμα. Οι έλεγχοι πρέπει να μπορούν να γίνουν μεμονωμένα είτε σε όλη την πλατφόρμα ως τελικός έλεγχος.

2.3 Τεχνική επισκόπηση AUTOSAR

Οι τεχνικοί στόχοι που έχει ορίσει το Autosar είναι η ευκολία μεταφοράς και συνεργασίας, μεταβλητότητα, ευχρηστία, ενσωμάτωση και η επεκτασιμότητα των λειτουργιών του λογισμικού που βασίζονται στο πρότυπο έτσι ώστε να χρησιμοποιούνται σε όλα τα επίπεδα αυτοματισμού των οχημάτων, κυρίως σε επίπεδο λογισμικού το οποίο κατ' επέκταση θέτει και τα όρια στην επιλογή του υλικού. Σε μια μονάδα ελέγχου (ECU) το Autosar προκαθορίζει τις τυποποιημένες διαδικασίες και διευκολύνει την παραμετροποίηση και την ανάπτυξη έτσι ώστε ο πελάτης να μπορέσει να φτάσει στο επιθυμητό αποτέλεσμα και στη συνέχεια να μπορέσει να πραγματοποιήσει ουσιαστικούς ελέγχους πάνω σε αυτό τόσο για την ποιότητα των λειτουργιών του λογισμικού όσο και τα υλικά που έχει επιλέξει.

Παρακάτω αναφέρονται επιγραμματικά τα τεχνικά χαρακτηριστικά του προτύπου όπως παρουσιάζονται επισημά από το Autosar.⁴

- Ευελιξία στην προσαρμογή των στοιχείων του λογισμικού ώστε να εξυπηρετούν τις απαιτήσεις του κάθε τελικού χρήστη ξεχωριστά καθώς τελικά η κάθε μονάδα είναι διαφορετική από την άλλη. Ο τρόπος που δημιουργείται είναι ο ίδιος.
- Επεκτασιμότητα του λογισμικού με στόχο να προσαρμόζεται στις ανάγκες της κάθε μονάδας επιτρέποντας ταυτόχρονα την αμφίδρομη επικοινωνία με τα άλλα συστήματα που πρέπει να συνεργάζεται.
- Δυνατότητα μεταφοράς των λειτουργιών του λογισμικού σε πολλά επίπεδα και συστήματα ενός οχήματος.
- Ευχρηστία και επαναχρησιμοποίηση των λειτουργιών του λογισμικού έτσι ώστε οι κατασκευαστές να έχουν τα εργαλεία για να μπορούν να βελτιώσουν την γραμμή παραγωγής τους και να αυξήσουν την ανταγωνιστικότητα της εταιρίας τους.
- Πρότυπες διεπαφές και επίπεδα λογισμικού δημιουργημένα με τυποποιημένη μέθοδο με στόχο την εύκολη συνεργασία μεταξύ προμηθευτών και κατασκευαστών.
- Στοιχεία λογισμικού, AUTOSAR (Software Components). Τα στοιχεία του λογισμικού είναι και το πιο σημαντικό κομμάτι του προτύπου. Συνεπώς πρέπει ο ορισμός τους, η μορφή τους και τα χαρακτηριστικά τους να είναι εξαιρετικά δομημένα για να μπορεί το λογισμικό να είναι λειτουργικό και ευέλικτο σε πολλά επίπεδα και σε πολλούς χρήστες.
- Εικονικός Δίαυλος επικοινωνίας (Virtual Functional Bus). Σε επίπεδο υλικού η επικοινωνία των συστημάτων στα οχήματα όπως έχει προαναφερθεί παραπάνω γίνεται με μέσω του Canbus το οποίο έχει υιοθετηθεί στην αυτοκινητοβιομηχανία τα τελευταία 20 χρόνια από όλους τους κατασκευαστές. Συνεπώς σε επίπεδο ανάπτυξης λογισμικού χρειάζεται η δημιουργία ενός εικονικού δίαυλου επικοινωνίας που θα προσομοιώνει την πραγματική λειτουργία αποτελεσματικά. Ο εικονικός δίαυλος είναι απαραίτητος για την ομαλή εξέλιξη του ελέγχου του λογισμικού.
- Περιορισμοί και περιγραφές για την διαμόρφωση των μονάδων ελέγχου με στόχο την ανάπτυξη μονάδων ελέγχου που είναι εξελίξιμες και απόλυτα συνεργάσιμες μεταξύ τους. Αξίζει να σημειώσουμε ότι τα στοιχεία του λογισμικού είναι ανεξάρτητα.
- Ορισμός και περιγραφή της μεθοδολογίας που πρέπει να χαρτογραφηθούν οι μονάδες ελέγχου σύμφωνα με το Autosar. Το αποτέλεσμα είναι η τελική διαμόρφωση και παραγωγή του βασικού λογισμικού και του RTE για την μονάδα ελέγχου.
- Παραγωγή του βασικού λογισμικού και του RTE για την κάθε μονάδα ελέγχου που είναι και ο βασικός σκοπός ύπαρξης του Autosar.

⁴ <http://www.autosar.org/about/technical-overview/>

2.3.1. Ανάπτυξη Autosar software components

Το Autosar βασίζεται στην ανάπτυξη βασικών μοντέλων για τα στοιχεία του λογισμικού που είναι και ο καθιερωμένος τρόπος προγραμματισμού στην αυτοκινητοβιομηχανία. Όταν μια εταιρία αποφασίσει να παράγει τον κώδικα για την εφαρμογή του στις μονάδες ελέγχου που κατασκευάζει, η ομάδα ανάπτυξης του λογισμικού δεν χρειάζεται να αλλάξει τη μεθοδολογία της, αρκεί να παραμετροποιήσει τα software components του Autosar και τα εργαλεία θα παράγουν τον τελικό κώδικα σε αρχείο τύπου .arxml που θα ενσωματωθεί στην μονάδα ελέγχου. Ο συγκεκριμένος τύπος αρχείου είναι ένα Autosar XML αρχείο, το οποίο δημιουργήθηκε για τις ανάγκες του προτύπου και δεν είναι τίποτα περισσότερο από ένα αρχείο που είναι ρυθμισμένο εξαρχής σύμφωνα με το πρότυπο.

Το κάθε εξάρτημα του λογισμικού είναι αναπόσπαστο μέρος της τελικής εφαρμογής και ισότιμο μέλος για την εύρυθμη λειτουργία της. Ανάλογα με τις απαιτήσεις το κάθε εξάρτημα είναι, μπορεί να είναι είτε μεγάλο μπλοκ που ουσιαστικά να αποτελεί υποσύστημα είτε να είναι μια απλή λογική συνάρτηση. Επιπλέον τα SW είναι αυτόνομα και μοναδικά για την κάθε μονάδα. Η διαδικασία δημιουργίας εξαρτημάτων χωρίζεται σε δυο μέρη, στον σχεδιασμό και εφαρμογή τα οποία παράγουν τον τελικό κώδικα που θα χρησιμοποιηθεί σε αρχεία μορφής .xml.

Αναλύοντας τα εξαρτήματα του λογισμικού του προτύπου προκύπτουν οι τύποι των SW που επικοινωνούν μέσω των πυλών των SW, μεταξύ τους και με τις διεπαφές του σε όλο το χάρτη του λογισμικού. Οι πύλες του Autosar ονομάζονται PortPrototypes και χαρακτηρίζονται από τις διεπαφές των πυλών που ονομάζονται PortInterfaces. Οι αποδεκτές μέθοδοι επικοινωνίας εντός των εξαρτημάτων λογισμικού που υποστηρίζονται από το πρότυπο είναι⁵:

- Βασίζεται στα δεδομένα χρησιμοποιεί τη σχέση αποστολέα – παραλήπτη
- Βασίζεται στις λειτουργίες χρησιμοποιεί τη σχέση πελάτη – διακομιστή
- Βασίζεται στα μοντέλα με συνέπεια να δημιουργούνται ανεξάρτητα μοντέλα σε κάθε επίπεδο του λογισμικού.
- Έτοιμές και αδρανής λειτουργίες που εκτελούνται κατ' επιλογή.

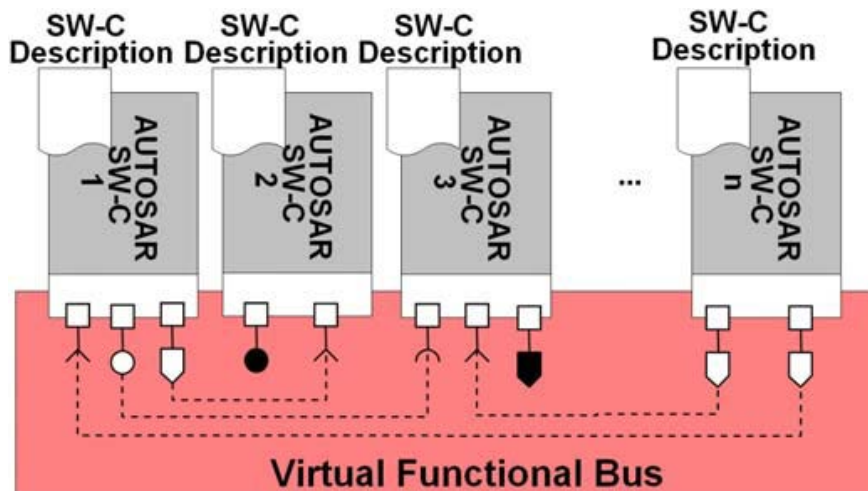
Σε συνέχεια, οι πύλες του μοντέλου AUTOSAR χρησιμοποιούνται αμφίδρομα με τις διεπαφές που επικοινωνούν. Δηλαδή υπάρχουν τρία διαφορετικά είδη πυλών που έχουν τη δυνατότητα να απαιτούν και να παρέχουν υπηρεσίες και δεδομένα. Η χρήση της κάθε πύλης ορίζεται ξεχωριστά από τον πελάτη.

Οι τύποι των εξαρτημάτων του λογισμικού υπάρχουν σε αφθονία. Παρόλα αυτά οι πιο διακεκριμένοι είναι δυο, ο ατομικός τύπος, που ονομάζεται AtomicSwComponentType από το μοντέλο, και ο πιο σύνθετος που διατίθεται και ονομάζεται CompositionSwComponentType. Ο σκοπός του CompositionSwComponentTypes είναι η συγκέντρωση των τύπων (SwComponentTypes) με σκοπό τη δημιουργία πιο περιπλοκών μονάδων περιπλοκότητας σε πιο θεωρητικό επίπεδο. Περιληπτικά παρόλο που τα CompositionSwComponentTypes επιτρέπουν την αφαίρεση ενός μέρους τους εσωτερικά σαν ένα ξεχωριστό σύστημα, είναι ένα αρχιτεκτονικό εργαλείο του προτύπου που επιτρέπει της επεκτασιμότητα της εφαρμογής. Τελικά ενώνουν εξαρτήματα λογισμικού που έχουν δημιουργηθεί ήδη βοηθώντας να σχεδιαστεί ευκολότερα το λογικό σχέδιο του λογισμικού. Αξίζει να σημειωθεί ότι δεν επηρεάζουν τον τρόπο όπου τύποι του λογισμικού αλληλεπιδρούν με το Virtual Functional Bus (VFB).

Η διαφορά του AtomicSwComponentType σε σχέση με το CompositionSwComponentType είναι σε θέση να περιέχει τους κανόνες και την εσωτερική δομή των τύπων των εξαρτημάτων του λογισμικού. Δηλαδή τα εσωτερικά δεδομένα και τα εκτελεστικά σύνολα (RunnableEntities), καθώς και τους συνδέσμους επικοινωνίας στο εσωτερικό των AtomicSwComponentTypes. Τελικά η ατομικοί τύποι είναι ουσιαστικά η πραγματική εφαρμογή του λογισμικού. Αξίζει να σημειωθεί ότι το Autosar δεν τυποποιεί και δεν περιορίζει τους τύπους ως προς την δομή και την επικοινωνία.

⁵ <http://www.autosar.org/about/technical-overview/software-component/>

2.3.2.Virtual Functional Bus

Εικόνα 3 VIRTUAL FUNCTIONAL BUS⁶

Το AUTOSAR καθορίζει το πρότυπο ενός επιπέδου του λογισμικού τόσο την αρχιτεκτονική όσο και την γλώσσα περιγραφής των εξαρτημάτων έτσι ώστε αυτά τα μέρη του λογισμικού να μπορούν να εφαρμοστούν ανεξάρτητα από το υλικό που θα έχει εγκατασταθεί, δηλαδή πολύ απλά, λογισμικό που «τρέχει» σε εικονικό υλικό αλλά εφαρμόζεται σε οποιοδήποτε πραγματικό υλικό. Με αυτό τον τρόπο επιτυγχάνεται ο έλεγχος και η απόδοση του λογισμικού σε ένα εικονικό λειτουργικό σύστημα AUTOSAR και επιβεβαιώνει την αποδοτική της επικοινωνία μεταξύ των λογισμικών εξαρτημάτων. Με αυτόν τον τρόπο οι κατασκευαστές απλοποιούν την σχεδίαση και ανάπτυξη των συστημάτων τους από τον σχεδιασμό του λογισμικού έχοντας και τη δυνατότητα ελέγχου σε υπολογιστικό περιβάλλον. Συγκριτικά με τις παραδοσιακές μεθόδους, το AUTOSAR όχι μόνο απλοποιεί την ανταλλαγή των εξαρτημάτων του λογισμικού αλλά επιπρόσθετα παρέχει μεθοδολογία για να χειριστεί και να διαχειριστεί την αυξημένη πολυπλοκότητα των συστημάτων των οχημάτων.

Ο εικονικός διάυλος επικοινωνίας (VFB) είναι η σύνδεση των λογισμικών συστημάτων ολόκληρου του οχήματος μεταξύ τους. Είναι δηλαδή το εικονικό CANBUS το οποίο είναι το πρωτόκολλο επικοινωνίας των πραγματικών συστημάτων στα οχήματα. Η επικοινωνία ανάμεσα στα διάφορα λογισμικά εξαρτήματα και συστήματα μπορεί να καθοριστεί ανεξάρτητα από οποιοδήποτε υλικό. Η λειτουργικότητα του VFB πραγματοποιείται από διαμορφωμένα πρότυπα επικοινωνίας.

Το AUTOSAR παρέχει υπηρεσίες και επικοινωνιακά πρωτόκολλα, βιβλιοθήκη προτύπων στο βασικό μέρος του λογισμικού. Ο στόχος είναι να παρέχει στον προγραμματιστή του λογισμικού ώστε να μπορέσει να αναπτύξει μια εφαρμογή που με πολύ μεγάλη λειτουργικότητα απαραίτητη προϋπόθεση για να επιτευχθεί η λειτουργικότητα είναι να όλα τα λογισμικά συστήματα AUTOSAR να τυποποιηθούν με βάση αυτό. Επιπλέον αξίζει να σημειωθεί ότι όταν οι υπηρεσίες AUTOSAR είναι τυποποιημένες τότε οι πύλες των εξαρτημάτων, τα προγράμματα οδήγησης του λογισμικού, και η αφαίρεση του ECU που είναι και μέρος του βασικού λογισμικού, είναι συγκεκριμένα.

Εξαρτήματα λογισμικού, πύλες και διεπαφές AUTOSAR

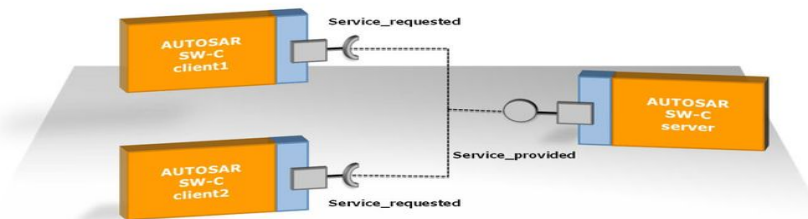
Το πρότυπο AUTOSAR δημιουργείται και αναπτύσσεται με βασικό συστατικό τα εξάρτημα του λογισμικού. Τα εξαρτήματα έχουν προκαθορισμένες και μοναδικές πόρτες δικτύου από τις οποίες έρχονται σε επαφή με άλλα εξαρτήματα λογισμικού. Η θεωρία των διεπαφών του AUTOSAR καθορίζει τις υπηρεσίες ή τις πληροφορίες οι οποίες απαιτούνται από μία πόρτα ενός εξαρτήματος. Οι πιο διαδεδομένες διεπαφές AUTOSAR που χρησιμοποιούνται είναι οι Client-Server διεπαφές οι οποίες περιέχουν έτοιμες ομάδες λειτουργιών και οι Sender-Receiver

⁶ Guido Sandmann, Richard Thompson, (2008), Development of AUTOSAR Software Components within Model-Based Design, The MathWorks, Inc

διεπαφές οι οποίες δια μέσου του VFB επιτρέπουν την ανταλλαγή πληροφοριών δεδομένων. Υπάρχουν και άλλοι τύποι διεπαφών λιγότερο σημαντικοί που επιτρέπουν την ανταλλαγή πληροφοριών, μεταβλητών ή σταθερών πληροφοριών ή ενεργοποιούν τις διαδικασίες.

Οι πύλες δικτύου των εξαρτημάτων είναι είτε τύπου PPort ή Rport όπου η PPort παρέχει μια διεπαφή, ενώ RPort προαπαιτεί μια. Στην περίπτωση επικοινωνίας Client-Server το εξάρτημα λογισμικού που ανήκει η πόρτα PPort (εμπεριέχει μια διεπαφή) παρέχει τις εργασίες που καθορίζονται από τη Client-Server διεπαφή. Στην περίπτωση της RPort τότε απαιτείται μια διεπαφή AUTOSAR, έτσι το εξάρτημα μπορεί να επικαλείται τις λειτουργίες όπως καθορίζονται μέσα στη διεπαφή του Client-Server. Στην περίπτωση επικοινωνίας Sender-Receiver το εξάρτημα του λογισμικού που ανήκει η Rport πύλη (εμπεριέχει μια διεπαφή), παράγει τα δεδομένα που περιγράφονται μέσα στη διεπαφή Sender-Receiver. Στην περίπτωση που η Rport πύλη ενός εξαρτήματος ζητήσει μία διασύνδεση του AUTOSAR, το εξάρτημα του λογισμικού λαμβάνει τις πληροφορίες που βρίσκονται μέσα σε μία διεπαφή του Sender-Receiver.

Επικοινωνία Client-Server

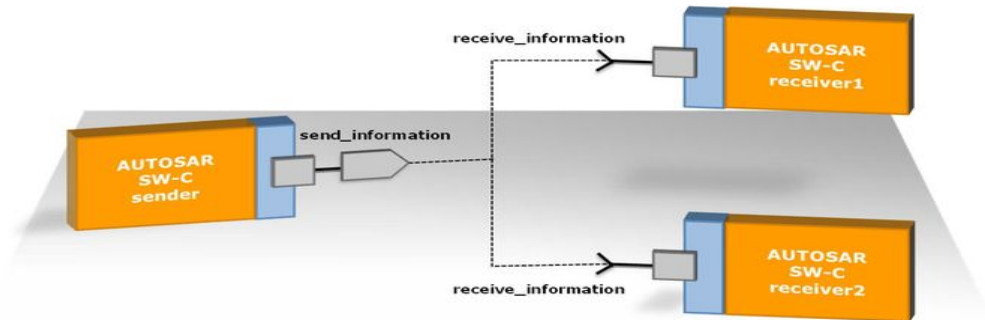


Εικόνα 4 Μοντέλο επικοινωνίας Client - Server⁷

Αρχικά ας δούμε το πιο γνώριμο και ευρέως γνωστό, καθώς χρησιμοποιείται σε όλες τις εφαρμογές, πρότυπο επικοινωνίας. Όπως μαρτυρά και ο τίτλος είναι μια τυπική σχέση δικτύου όπου ο εξυπηρετητής είναι αυτός που προμηθεύει την υπηρεσία και ο client είναι ο χρήστης της υπηρεσίας.

Ο client ξεκινάει την επικοινωνία ζητώντας από το διακομιστή να εκτελέσει μία λειτουργία μεταφέροντας τις απαραίτητες πληροφορίες. Ο διακομιστής περιμένει αιτήματα από ένα client, και στη συνέχεια εκτελεί τη απαιτούμενη υπηρεσία, και αποστέλλει μία ανταπόκριση το αιτήματος του client. Αναλόγως την κατεύθυνση της πληροφορίας το εξάρτημα του λογισμικού AUTOSAR είναι client ή server, ένα εξάρτημα μπορεί να είναι και τα client και server, το οποίο καθορίζεται από την κατασκευή του λογισμικού. Επιπλέον ο client μπορεί να μπλοκαριστεί αφού στείλει το αίτημα και τότε υπάρχει συγχρονισμένη επικοινωνία ή να μην μπλοκαριστεί τότε υπάρχει ασύγχρονη επικοινωνία. Όλες οι επικοινωνίες πραγματοποιούνται με το μοντέλο του VFB.

⁷ <https://www.autosar.org/about/technical-overview/virtual-functional-bus/communication/>

Επικοινωνία Sender-Receiver**Εικόνα 5 Μοντέλο επικοινωνίας Sender - Receiver⁸**

Η επικοινωνία αποστολέα-παραλήπτη εξυπηρετεί την διακίνηση της ασύγχρονης επικοινωνίας όπου ο αποστολέας διαμοιράζει τις πληροφορίες σε ένα ή και περισσότερους παραλήπτες. Ο αποστολέας δεν είναι μπλοκαρισμένος (ασύγχρονη επικοινωνία) και ούτε αναμένει ούτε λαμβάνει ανταπόκριση από τους παραλήπτες ο αποστολέας απλά δίνει τις πληροφορίες και οι παραλήπτες αποφασίζουν ανώνυμα πότε και πως θα χρησιμοποιήσουν τις πληροφορίες. Είναι ευθύνη του ίδιου του δικτύου της επικοινωνίας να διανέμει τις πληροφορίες. Ο αποστολέας δεν ξέρει ούτε ποιοι ούτε πόσοι είναι οι παραλήπτες.

2.3.3. Αρχιτεκτονική του λογισμικού ECU

Το λογισμικό του AUTOSAR το οποίο βρίσκεται στο τελευταίο επίπεδο του προτύπου εμπεριέχει το λογισμικό που είναι χαρτογραφημένο στην μονάδα ελέγχου (ECU). Στη βάση υπάρχει το υλικό της μονάδας ελέγχου (Hardware) από εξαρτήματα του λογισμικού του AUTOSAR όπου εμπεριέχονται οι διεπαφές που περιγράφηκαν παραπάνω. Όλη η αλληλεπίδραση ανάμεσα στα λογισμικά AUTOSAR εξαρτήματα και τα ατομικά λογισμικά εξαρτήματα γίνεται από το περιβάλλον λειτουργίας του AUTOSAR το οποίο βρίσκεται ανάμεσα στη βάση που είναι το υλικό και την κορυφή που είναι το λογισμικό.

AUTOSAR Runtime Environment

Σε επίπεδο σχεδιασμού ενός συστήματος το περιβάλλον ανάπτυξης του AUTOSAR είναι ένα επικοινωνιακό κέντρο για ανταλλαγή πληροφοριών μεταξύ εντός και εκτός της μονάδας ελέγχου ECU. Το RTE παρέχει την δυνατότητα επικοινωνίας των λογισμικών εξαρτημάτων του AUTOSAR που είναι συνδεδεμένα με αυτό με το να παρέχει ίδια διεπαφή και υπηρεσίες ανεξάρτητα αν παρεμβάλλονται αλλά πρότυπα επικοινωνίας στην ECU ή η επικοινωνία παραμένει σε εσωτερικά -ECU. Καθώς τα επικοινωνιακά κριτήρια των λογισμικών εξαρτημάτων που είναι υπεύθυνα για την λειτουργία της RTE χρειάζεται να προσαρμόζονται στην εκάστοτε ECU και στη διαμόρφωση της συνεπάγεται ότι το τελικό RTE θα διαφέρει από το ένα ECU στο άλλο.

⁸ <https://www.autosar.org/about/technical-overview/virtual-functional-bus/communication/>

Βασικό λογισμικό AUTOSAR

Το βασικό λογισμικό είναι το τυποποιημένο λογισμικό όπου υπάρχουν λογισμικά εξαρτήματα του προτύπου, και είναι αναγκαίο για το λειτουργικό μέρος του λογισμικού. Το βασικό λογισμικό περιλαμβάνει τυποποιημένες και συγκεκριμένες μονάδες λογισμικού που είναι όμως απαραίτητες για την λειτουργία της ECU σύμφωνα με το AUTOSAR. Για να υπάρξει μια πλήρως λειτουργική μονάδα προϋποθέτει λογισμικό της μονάδας ελέγχου από τον κατασκευαστή, συνεπώς για το βασικό λογισμικό δεν παρέχει κάποια αυτόνομη και πλήρως λειτουργική εργασία και εμπεριέχεται κάτω από το περιβάλλον λειτουργίας του AUTOSAR το οποίο όμως συμπεριλαμβάνει υπηρεσίες συστημάτων όπως διαγνωστικά πρωτόκολλα και διαχείριση NVRAM. Η Επικοινωνιακή βιβλιοθήκη περιλαμβάνει τα περισσότερα πρωτοκολλά που έχουνε εφαρμοστεί στο παρελθόν ή εφαρμόζονται ακόμα και σήμερα εμπεριέχοντας και δικτυακή διαχείριση.

Βασικό εξάρτημα του λογισμικού μιας μονάδας ελέγχου είναι διαίρεση της μονάδας ελέγχου έτσι ώστε να παρέχει στο λογισμικό μια διεπαφή που παρέχει τιμές για οποιοδήποτε υλικό εντός της ECU καθώς και δυνατότητα ενεργοποίησης και απενεργοποίησης αυτού. Ακόμα ένα βασικό εξάρτημα του λογισμικού είναι το σύνθετο πρόγραμμα οδήγησης (CDD) το οποίο επιτρέπει την απευθείας πρόσβαση στο υλικό όπου είναι σημαντικό και απαιτείται.

Λειτουργικό σύστημα

Καθώς η αρχιτεκτονική του AUTOSAR στοχεύει στο να είναι κοινή σε όλους τους τομείς ενός οχήματος, θα προσδιορίζει τις απαιτήσεις για ένα λειτουργικό σύστημα AUTOSAR. Παραδειγματικά το λειτουργικό σύστημα πρέπει να είναι διαμορφωμένο και να έχει κλιμακώνεται στατικά, να αποδίδει σε πραγματικό χρόνο, να μπορεί να δημιουργεί πρόγραμμα βάση προτεραιότητας και να προστατεύει την λειτουργία των διεργασιών.

Το πρότυπο επιτρέπει την ενσωμάτωση λειτουργικών συστημάτων στο βασικό λογισμικό του AUTOSAR λογισμικά εξαρτήματα. Για να δημιουργηθούν οι διεπαφές και να είναι συμβατές με το πρότυπο το ξένο λειτουργικό σύστημα πρέπει να αντληθεί σε ένα λειτουργικό σύστημα του AUTOSAR. Τέλος αξίζει να σημειωθεί ότι το πρότυπο λειτουργικό σύστημα OSEK (ISO 17356-3) χρησιμοποιείται σαν η βάση για το λειτουργικό σύστημα του AUTOSAR.

Το πρότυπο OSEK προκύπτει από τη συντομογραφία του Open Systems and the Corresponding Interfaces For Automotive Electronics. Αποτελείται από το λειτουργικό σύστημα του, τις επικοινωνίες και τη διαχείριση δικτύου του. Το συγκεκριμένο πρότυπο παρέχει αποτελεσματικό προγραμματισμό για διεργασίες, διεργασίες διακοπής, ενεργοποίηση διεργασιών με βάση το χρόνο και υποστήριξη debugging. Αντίθετα δεν παρέχει προγράμματα οδήγησης για δίκτυα, αρχεία και γραφικού περιβάλλοντος. Επιπλέον δεν παράγει διεργασίες, δεν υποστηρίζει επεξεργαστές πολλών πυρήνων και δεν προστατεύει τη μνήμη. Εν κατακλείδι είναι ένα πρότυπο το οποίο έχει χρησιμοποιηθεί ευρέως σε πολλά τμήματα των οχημάτων και σαν βάση είναι αρκετά ικανό και αξιόπιστο.

Microcontroller Abstraction

Το Microcontroller Abstraction Layer (MCAL) είναι ξεχωριστό επίπεδο στο πρότυπο και χρησιμοποιείται για την διαχείριση των μικροελεγκτών δίνοντας έτοιμες μονάδες στα υψηλότερα επίπεδα λογισμικού. Το MCAL είναι συγκεκριμένο επίπεδο που αφορά το υλικό της μονάδας ελέγχου. Κυρίως στόχος του είναι να εξασφαλίζει μία σταθερές διεπαφές με τα εξαρτήματα του βασικού λογισμικού. Διαχειρίζεται τους περιφερειακούς μικροελεγκτές και παρέχει στα εξαρτήματα του βασικού λογισμικού με τις πληροφορίες που χρειάζονται όσον αφορά τους μικροελεγκτές. Το MCAL παρέχει πληροφορίες με τη μορφή ειδοποιήσεων για όλες τις διεργασίες των μικροελεγκτών όπως οι παρακάτω⁹:

- Digital I/O (DIO)
- Analog/Digital Converter (ADC)
- Pulse Width Modulator (PWM)
- EEPROM (EEP)
- Flash (FLS)

⁹ <http://www.autosar.org/about/technical-overview/ecu-software-architecture/autosar-basic-software/>

- Output Compare Unit (OCU)
- Watchdog Timer (WDT)
- Serial Peripheral Interface (SPI)
- I2C Bus (IIC)

Διεπαφές AUTOSAR

Το πρότυπο AUTOSAR παρέχει τυποποιημένα διάγραμμα για την δημιουργία διεπαφών με βάση αυτό από οποιαδήποτε εφαρμογή. Υπάρχουν τρεις διαφορετικοί τύποι διασυνδέσεων η "AUTOSAR Interface (διασύνδεση)", η "Standardized AUTOSAR Interface (τυποποιημένη διασύνδεση AUTOSAR)" και η "Standardized Interface" (τυποποιημένη διασύνδεση).

Μία διεπαφή AUTOSAR αφορά μόνο τα δεδομένα και τις υπηρεσίες που απαιτούνται ή παρέχονται από ένα εξάρτημα λογισμικού, καθορίζεται και εφαρμόζεται με βάση τους ορισμούς της διεπαφής AUTOSAR. Ο καθορισμός της διεπαφής όμως μπορεί να είναι και μερικώς καθώς υπάρχει η δυνατότητα να συμπεριλαμβάνει συγκεκριμένες απαιτήσεις του κατασκευαστή. Η διεπαφή επιτρέπει σε διαφορά λογισμικά εξαρτήματα να αλληλοεπιδρούν μεταξύ των μονάδων ελέγχου μέσω του περιβάλλοντος επικοινωνίας (RTE) που λειτουργούν στις ECU.

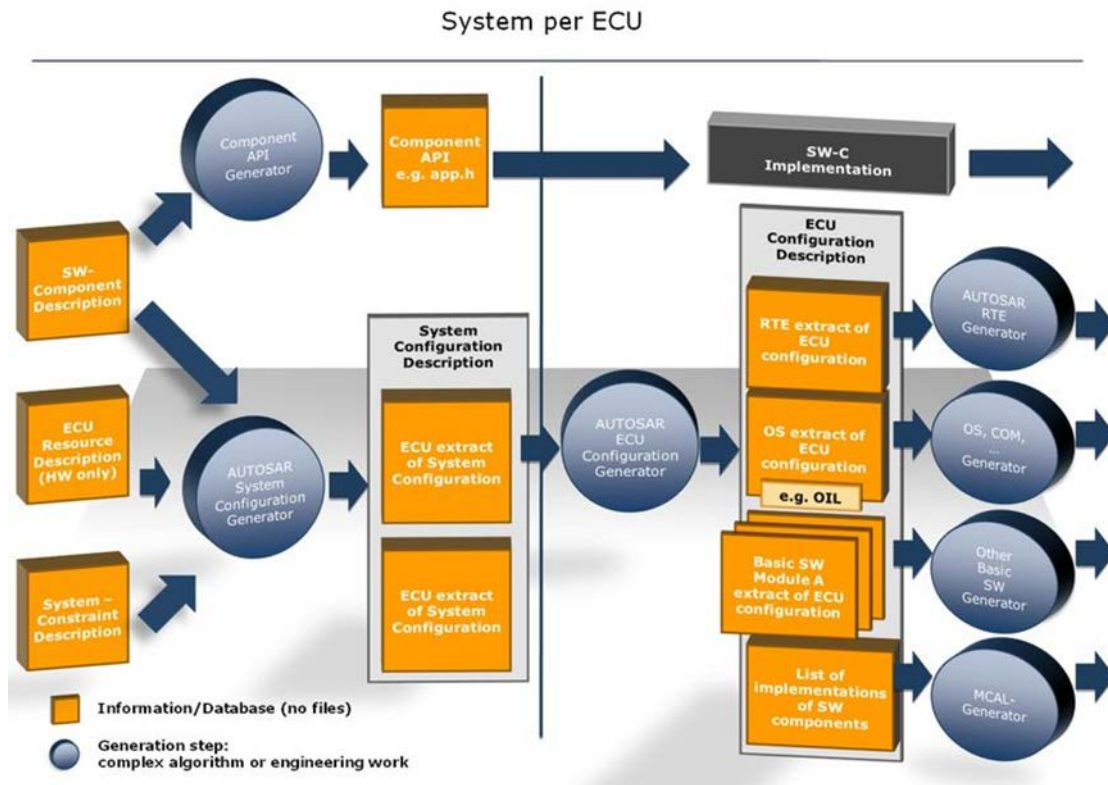
Τυποποιημένη διεπαφή AUTOSAR.

Μια τυποποιημένη διεπαφή AUTOSAR είναι διεπαφή που είναι πλήρως τυποποιημένη από το AUTOSAR πρότυπο. Είτε παρέχει στα εξαρτήματα του λογισμικού ολοκληρωμένες διεπαφές με βάση το πρότυπο και εμπεριέχονται στο βασικό λογισμικό του προτύπου είτε παράγονται εσωτερικά του AUTOSAR.

Τυποποιημένη διεπαφή

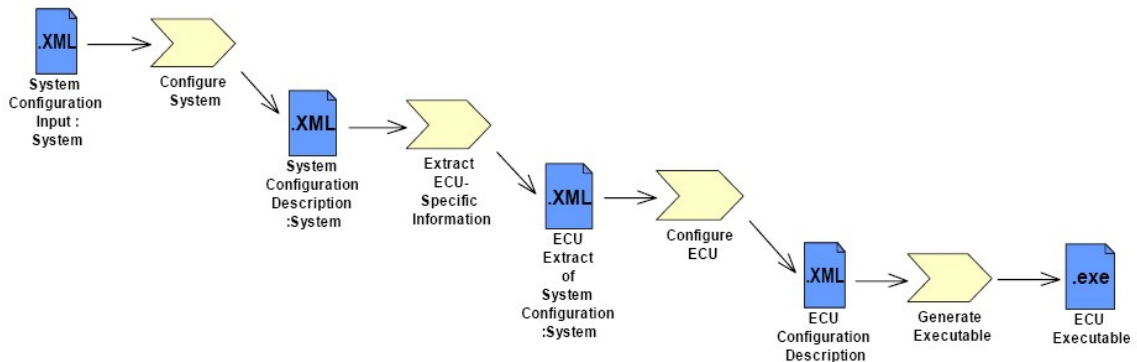
Μια τυποποιημένη διεπαφή ορίζεται όταν υπάρχει τυποποίηση API.

2.3.4. Η μεθοδολογία του AUTOSAR



Εικόνα 6 Μεθοδολογία AUTOSAR¹⁰

Αρχικά, η μεθοδολογία του AUTOSAR ορίζεται η τεχνική για την ανάπτυξη συστημάτων με βάση αυτό. Στη μεθοδολογία, λοιπόν, παρουσιάζονται όλα τα στάδια από την σχεδίαση του συστήματος έως και τους δοκιμαστικούς ελέγχους του τελικού εκτελέσιμου συστήματος, χωρίς να είναι απολυτά τα βήματα που περιγράφονται, ούτε και η σειρά τους. Είναι μία πιθανή ροή πληροφοριών δηλαδή «οδηγίες» ανάπτυξης αλλά σε θεωρητικό επίπεδο καθώς δεν είναι ολοκληρωμένες καθώς στόχος του AUTOSAR είναι να παρέχει ελευθέρια κινήσεων στον τελικό χρήστη.



Εικόνα 7 Τα στάδια της μεθοδολογίας από την διαμόρφωση του συστήματος στο εκτελέσιμο αρχείο.¹¹

¹⁰ <https://www.autosar.org/about/technical-overview/autosar-methodology/>

¹¹ https://retis.sssup.it/sites/default/files/lesson19_autosar.pdf, page 83

Αρχικά πρέπει να οριστεί η διαμόρφωση του συστήματος που θα εισαχθεί, τα εξαρτήματα του λογισμικού, το υλικό και οι περιορισμοί του συστήματος. Το AUTOSAR παρέχει την πρότυπη μορφή που πρέπει να υπάρχουν οι παραπάνω πληροφορίες. Πιο συγκεκριμένα:

- Κάθε εξάρτημα του λογισμικού απαιτείται να παρέχει τους τύπους των δεδομένων, τις πύλες του και τις διεπαφές του.
- Οι πόροι και οι προδιαγραφές κάθε μονάδας ελέγχου που απαιτούνται. Τα πιο σημαντικά είναι ο επεξεργαστής, η μνήμη, οι ενεργοποιητές και οι αισθητήρες.
- Οι περιορισμοί του συστήματος όπως σήματα επικοινωνίας και χαρτογράφηση των λογισμικών εξαρτημάτων.

Η διαμόρφωση του συστήματος είναι μια διαδικασία που ποικίλει ανάλογα με το τη μονάδα ελέγχου που θα χρησιμοποιηθεί διότι κάποια εξαρτήματα δεν μπορούν να χαρτογραφηθούν σε κάποιες μονάδες και αντίστοιχα κάποιες μονάδες απαιτούν συγκεκριμένα εξαρτήματα. Για να γίνει η χαρτογράφηση και διαμόρφωση του συστήματος σύμφωνα με το AUTOSAR απαιτείται μηχανολογική γνώση και αρίστη γνώση του υλικού. Ωστόσο το πρότυπο παρέχει την εργαλειοθήκη που είναι απαραίτητη για την διαμόρφωση του συστήματος. Το τελικό αποτέλεσμα της διαμόρφωσης του συστήματος είναι το System Configuration Description το οποίο περιέχει την τοπολογία δικτύου και την χαρτογράφηση των εξαρτημάτων του λογισμικού των μονάδων ελέγχου. Παρακάτω εξετάζονται οι υπόλοιποι ορισμοί όπως παρουσιάζονται στο διάγραμμα της μεθοδολογίας του AUTOSAR.

- **Extract ECU**: είναι οι πληροφορίες από τη διαμόρφωση του συστήματος που απαιτούνται για μια συγκεκριμένη ECU.
- **System Configuration Extractor**: είναι η λειτουργία που αντλεί τις πληροφορίες διαμόρφωση του συστήματος που χρειάζονται για μια συγκεκριμένη μονάδα ελέγχου (ECU).
- **ECU Configuration Description**: αφορά κυρίως τη διαμόρφωση του περιβάλλοντος επικοινωνίας της ECU και των μονάδων της, και εμπεριέχει όλες τις πληροφορίες από την διαμόρφωση του βασικού λογισμικού και είναι τοπικές για μια συγκεκριμένη μονάδα ελέγχου. Το εκτελέσιμο λογισμικό, ο κώδικας από κομμάτια του βασικού λογισμικού και των λογισμικών εξαρτημάτων μπορεί να δημιουργηθεί από αυτές τις πληροφορίες.

2.3.5. Δοκιμαστικοί έλεγχοι

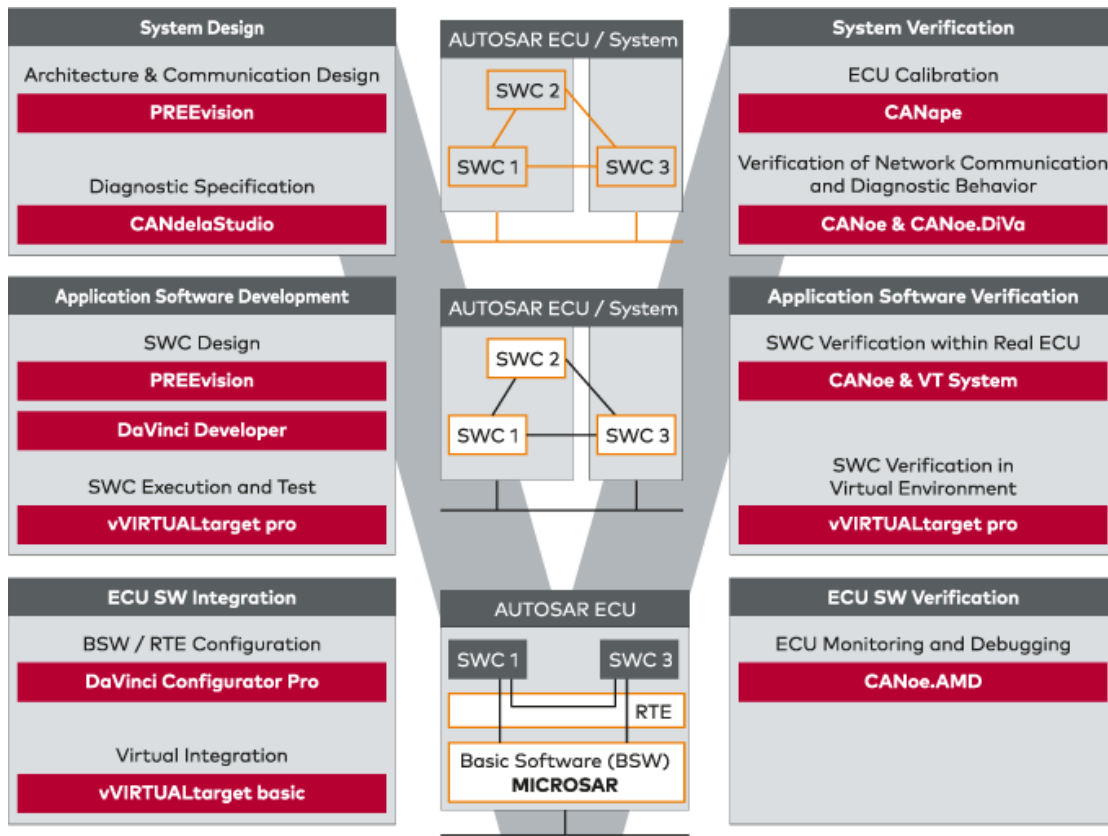
Σημαντικό κομμάτι του AUTOSAR είναι η δυνατότητα δοκιμαστικών ελέγχων σε όλα τα επίπεδα της εφαρμογής καθώς η δυνατότητα προσομοίωσης, εκτέλεσης και τελικά ελέγχου της εφαρμογής σε εργαστηριακό επίπεδο κρίνεται απολύτως αναγκαία για να παραμένει το AUTOSAR ανταγωνιστικό σε επίπεδο παραγωγής και οικονομίας, καθώς η δυνατότητα δοκιμής μειώνει το κόστος και την ταχύτητα παραγωγής μιας μονάδας ελέγχου και βελτιώνει την επεκτασιμότητα και την συντήρηση της. Επιπλέον παρέχει στον προγραμματιστή την απαραίτητη ευελιξία για την σωστή ανάπτυξη της εφαρμογής.

Πιο συγκεκριμένα μπορεί να εξεταστεί η συμβατικότητα της εφαρμογή δηλαδή τα εξαρτήματα του λογισμικού, το περιβάλλον επικοινωνίας και οι βιβλιοθήκες του λογισμικού που έχουν δημιουργηθεί. Ακόμα ένα σημαντικό στοιχείο που δοκιμάζει το πρότυπο είναι το δίκτυο που έχει δημιουργηθεί όπως τα πρωτόκολλα επικοινωνίας και οι συμπεριφορά του δικτύου.

2.3.6. Επιλογή Εργαλείων λειτουργίας και συντήρησης του συστήματος Autosar

Για μια επιτυχής εφαρμογή όλων των μεθόδων του Autosar υπάρχουν αρκετές επιλογές από εργαλεία, τα οποία μπορούν να χρησιμοποιηθούν είτε το καθένα μόνο του ή ακόμη και συνδυαστικά. Αξίζει να σημειωθεί ότι όλα τα παρακάτω εργαλεία δεν είναι ελεύθερα και απευθύνονται είτε σε βιομηχανίες είτε σε εταιρίες που κατασκευάζουν εξαρτήματα για τους κατασκευαστές οχημάτων. Τα κυριότερα και πιο ολοκληρωμένα από αυτά είναι τα ακόλουθα:

Vector

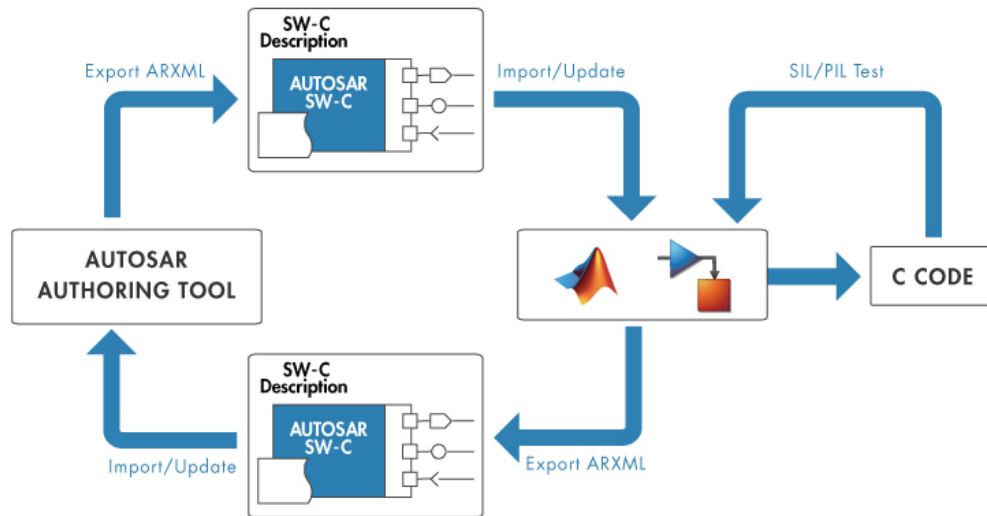
Εικόνα 8 Εργαλεία Vector¹²

Μία από τις κυρίαρχες εταιρείες στο τομέα ανάπτυξης συστημάτων της Αυτοκινητοβιομηχανίας είναι η Vector. Έχοντας εμπειρία στο χώρο εικοσιπέντε χρόνια έχει καταφέρει να προσφέρει μία ολοκληρωμένη πλατφόρμα Autosar καλύπτοντας έτσι όλο το φάσμα του. Η Vector διαθέτει εργαλεία, βασικό λογισμικό (Microsar), υπηρεσίες μηχανικής, επιτόπιου υποστήριξη καθώς και μαθήματα κατάρτισης εφόσον απαιτείται. Επιπρόσθετα παρέχει άλλα τόσα εργαλεία για έλεγχο της μονάδας διαχείρισης του κινητήρα.

Μέσω του βασικού λογισμικού για Autosar, Microsar, επιτυγχάνεται μια αποτελεσματική λύση μονάδας διαχείρισης του κινητήρα. Το Microsar χωρίζεται σε δύο μέρη, το Microsar RTE το οποίο ελέγχει το περιβάλλον εργασίας και το Microsar (BSW) το οποίο καλύπτει όλες τις πτυχές του πρότυπου Autosar. Επιπρόσθετα υπάρχουν περαιτέρω τα εργαλεία Da Vinci για τα γραφικά του Autosar αλλά και το Da Vinci Configurator Pro που χρησιμοποιείται για τη διαμόρφωση του βασικού λογισμικού Microsar την ώρα λειτουργίας του. Ένα από τα σημαντικότερα πλεονεκτήματα του Microsar είναι ότι είναι ένα λογισμικό το οποίο παραδίδεται με το «κλειδί στο χέρι» μιας και είναι προ ρυθμισμένο να εξυπηρετεί συγκεκριμένες OEM λειτουργίες. Ακόμα ένα θετικό στοιχείο του, είναι η απλοϊκότητά του καθώς παρέχει μία ολοκληρωμένη λύση Autosar από μόνο μία πηγή κώδικα και επίσης διατίθεται σε πολλές διαφορετικές πλατφόρμες υλικού (hardware platforms) και compilers. Η παράδοση και εγκατάσταση του λογισμικού γίνεται με ένα πλήρες βοηθητικό εγχειρίδιο.

¹² https://vector.com/vi_autosar_tools_en.html

Mathworks



Εικόνα 9 Εργαλεία Mathworks¹³

Τα Mathworks σε συνεργασία με τους κατασκευαστές, προμηθευτές και προγραμματιστές των εργαλείων του Autosar συμβάλλουν στην ανάπτυξη και βελτίωση των συστημάτων. Σαν συνεργάτης, διαθέτει ενσωματωμένο πακέτο υποστήριξης για προγραμματισμό του Autosar με κύριες εφαρμογές του τις Simulink και Stateflow. Και οι δύο αυτές εφαρμογές συμβάλουν δραστικά στο σχεδιασμό λογισμικού, με το Simulink σε συνδυασμό με το ενσωματωμένο πακέτο στήριξης να επιτυγχάνεται μία σωστή ένωση μοντέλων του Simulink με άλλα στοιχεία του Autosar.

Επιπλέον, τα Mathworks είναι ευέλικτα, για ένα καλύτερο αποτέλεσμα, να συνδυαστούν με άλλα προγράμματα όπως τα Vector Informatik DaVinci Developer, KPIT Cummins K-SAR, Mentor Graphics Volcano Vehicle System Architect και ETAS ISOLAR-A. Όπως παρουσιάζεται στην εικόνα πιο πάνω, βλέπουμε ότι η Matlab έχει καθοριστικό ρόλο σε όλη τη διάρκεια δημιουργίας του λογισμικού Autosar. Το ενσωματωμένο πακέτο χρησιμοποιείται αρχικά μέσω κώδικα γραμμένο στη C (C code). Έπειτα, από τα Mathworks αποστέλλεται ο κώδικας στο μέρος του λογισμικού που ονομάζεται Autosar Software components (Autosar SW-C) για να ενσωματωθεί στην εφαρμογή που τρέχει στην υποδομή του Autosar.

Τα Mathworks έχουν σαν κύριο λογισμικό τους για το πρόγραμμα Autosar το “Embedded Coder”. Το συγκεκριμένο εργαλείο έχει ένα ολοκληρωμένο πακέτο στήριξης το οποίο με την εγκατάστασή του προσφέρει πληθώρα επιλογών στο Simulink. Για παράδειγμα δύναται η δημιουργία και η τροποποίηση μίας σύνθεσης Autosar για ένα μοντέλο. Επίσης ο χρήστης έχει την δυνατότητα να φτιάξει κάποια στοιχεία, κομμάτια, για το μοντέλο Autosar.

Πολύ βασικό στη χρήση του “Embedded Coder” για τη δημιουργία μοντέλων Simulink είναι η σωστή κατανόηση του σκοπού της εφαρμογής που είναι η αποτελεσματικότητα και η σωστή παρακολούθηση της παραγωγής κώδικα σε όλα τα στάδια. Εκτός από αυτό, πολύ σημαντικό όταν δημιουργείται ένα μοντέλο Simulink να λαμβάνεται υπόψη τα μοντελοποιημένα πρότυπα για τον C κώδικα.

Arc Core

Η Arc Core είναι ένας ακόμη σημαντικός παροχέας του Autosar, προσφέροντας μια ολοκληρωμένη εργαλειοθήκη από προϊόντα για άμεσες λύσεις από ενσωματωμένα συστήματα

¹³ <https://www.mathworks.com/solutions/automotive/standards/autosar.html>

στο χώρο της αυτοκινητοβιομηχανίας. Βασιζόμενοι σε χαμηλές τιμές, η Arc Core ανέπτυξε τρεις εφαρμογές οι οποίες προσφέρουν άμεσα αποτελέσματα και είναι οι ακόλουθες:

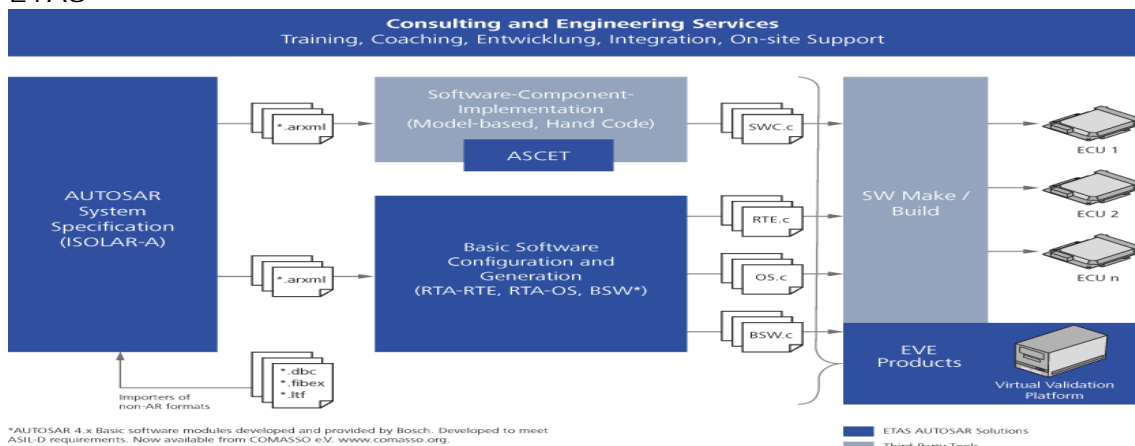
- Arctic Core: Μία ενσωματωμένη πλατφόρμα η οποία συμπεριλαμβάνει μία ολοκληρωμένη σειρά από χαρακτηριστικά που χρειάζονται για τη δημιουργία μιας ηλεκτρονικής μονάδας στη βιομηχανία αυτοκινήτου. Η πλατφόρμα υποστηρίζει πλήρως το Autosar με ενσωματωμένο κώδικα για πλήρη ευελιξία και διαφάνεια.
- Arctic Studio: Μια σειρά από εργαλεία για ανάπτυξη των μελετών του Autosar. Η εργαλειοθήκη μπορεί να στηρίξει σε όλα τα στάδια τη δημιουργία του πρότυπου Autosar δίνοντας λύσεις σε διαφορετικές φάσεις όπως στην ανάπτυξη της εφαρμογής, στην ανάπτυξη της ενσωματωμένης πλατφόρμας και στην αναβάθμιση του συστήματος.
- Arctic Bootloader: Ένα βοηθητικό ενσωματωμένο πρόγραμμα εκκίνησης για διευκόλυνση της αυστηρά ελεγχόμενης αλλαγής του λογισμικού που χρησιμοποιείται την ώρα που τρέχει στο ECU. Η χρήση ενός bootloader επιτρέπει τη λήψη νέων εικόνων λογισμικού και την ελεγχόμενη εκκίνηση της νέας εικόνας.

Τα πλεονεκτήματα μέσω των τριών εφαρμογών είναι ότι ο χρήστης καταφέρνει να αυξήσει την παραγωγικότητα του λογισμικού, να βελτιώσει την ποιότητά του και παράλληλα να κάνει διαχειρίσιμη την πολυπλοκότητα. Έχοντας πλήρη πρόσβαση σε ενσωματωμένο κώδικα και εργαλεία από ανοιχτές πηγές εργασιών γίνεται ευέλικτο και υπάρχει διαφάνεια.

Mentor

Η Mentor σε συνεργασία με τη Siemens προσφέρουν επίσης μία ολοκληρωμένη σειρά προϊόντων σχεδιασμού AUTOSAR στην αυτοκινητοβιομηχανία, Volcano. Είναι ένα σύνολο εργαλείων σχεδιασμού συστημάτων οχημάτων VSx τα οποία επιτρέπουν στους χρήστες να δημιουργήσουν το σύστημα οχημάτων AUTOSAR. Η σειρά συμπεριλαμβάνει ενσωματωμένο σχεδιασμό λογισμικού με τη δομή του βασισμένη στην Eclipse. Επίσης συμπεριλαμβάνει εργαλεία για αρχιτεκτονικό σχεδιασμό (Volcano Vehicle System Architect (VSA)) καθώς και εργαλεία ανάπτυξης λογισμικού εφαρμογών, εικονικής επικύρωσης και δοκιμής λογισμικού (Volcano VSA COM Designer). Τα εργαλεία αυτά βελτιώνουν το χρόνο διάθεσης στην αγορά και προσφέρουν πλεονεκτήματα κόστους επιτρέποντας τη χρήση τυποποιημένων διεπαφών και εξαρτημάτων βάσει των προδιαγραφών του AUTOSAR. Η Mentor μέσω της σειράς Volcano (VSA), έχει κατορθώσει μία πραγματική συμμόρφωση με τους κανόνες και τις απαιτήσεις του AUTOSAR. Η VSA στηρίζει ένα ολοκληρωμένο μοντέλο AUTOSAR συμπεριλαμβάνοντας επιπρόσθετα όλες τις μορφές του. Έχοντας την Eclipse, η οποία ανοιχτή και ευέλικτη, πλατφόρμα, επιτρέπει τη δημιουργία περιβάλλοντος σχεδιασμού προσαρμοσμένου στις ανάγκες του πελάτη.

ETAS



Εικόνα 10 Εργαλεία ETAS¹⁴

Σε μία προσπάθεια στήριξης της ιδέας του Autosar περί ενιαίου λογισμικού σχεδιασμού, αρχιτεκτονικής στην αυτοκινητοβιομηχανία, η Etas παρέχει στους πελάτες την υποστήριξη και

¹⁴ https://www.etas.com/en/products/applications_autosar.php

την εισαγωγή στο πρότυπο AUTOSAR. Σε συνεργασία με τη Commasso (ομοσπονδία του λογισμικού Autosar) προσφέρει ενσωματωμένο λογισμικό, μια ολοκληρωμένη εργαλειοθήκη και υπηρεσίες στήριξης.

Τα ETAS Autosar tools, σαν ολοκληρωμένο χαρτοφυλάκιο εργαλείων, δίνει πολλές επιλογές για ανάπτυξη του βασικού λογισμικού. Μερικά εργαλεία είναι τα ISOLAR-A, κώδικας στη C (C-code) για παραγωγή των κεφαλαίων COMASSO BSW καθώς και το BCT σαν πρόσθετο του ISOLAR.

Οι λύσεις ETAS έχουν σχεδιαστεί για να επιτρέπουν την ομαλή διαδικασία ανάπτυξης AUTOSAR με πρωτοπόρες τεχνολογίες, όπως το ETAS ISOLAR-EVE, για τη δοκιμή των εξαρτημάτων AUTOSAR στον υπολογιστή ή με άλλα ελεγμένα εργαλεία ανάπτυξης, όπως τα ETAS ASCET, ETAS INTECRIO, ETAS ISOLAR- A ή ETAS INCA.

Κατά τη διαδικασία λειτουργίας του λογισμικού (ETAS Autosar) το πλεονέκτημα είναι ότι επιτρέπεται πολύ εύκολα η αναβάθμιση του σε ήδη υφιστάμενο αναπτυγμένο περιβάλλον. Επιπλέον η ETAS, χρησιμοποιώντας την δικιά της τεχνογνωσία για το ενσωματωμένο λογισμικό, κατάφερε να προσφέρει τις υπηρεσίες και τα εργαλεία για σωστή διαμόρφωση και αναβάθμιση της εφαρμογής και του βασικού λογισμικού από οποιαδήποτε πηγή προέλευσης; για παράδειγμα είτε προέρχεται από ένα εξωτερικό MCAL (Microcontroller Abstraction Layer) ή από ανάπτυξη συγκεκριμένου (custome-specific) CDD (Complex Device Driver).

dSpace

Η dSPACE, έχοντας σχεδόν 30 χρόνια εμπειρία στο χώρο των συστημάτων, έδειξε θερμή υποστήριξη στην ανάπτυξη και τη γενική χρήση του προτύπου AUTOSAR. Παράλληλα ενέπνευσε πολλούς πελάτες να συνεργαστούν με τα εργαλεία υποστήριξης του AUTOSAR για περαιτέρω βελτίωση και πολλά επιτυχημένα έργα και καινοτομίες. Το πρότυπο AUTOSAR έχει εισχωρήσει σε όλους τους παραδοσιακούς τομείς οχημάτων, από τα ηλεκτρονικά οχήματα, έως τη δυναμική του οχήματος. Η αλυσίδα εργαλείων SystemDesk, TargetLink, TargetLink and SystemDesk, VEOS, Bus Implementation Tools, Bus Manager της dSPACE υποστηρίζουν τις συνήθειες εξελίξεις AUTOSAR σε όλα τα στάδια.

Ξεκινώντας από το SystemDesk, ο χρήστης έχει πρόσβαση στο σχεδιασμό, στην υλοποίηση και ενοποίηση σύνθετων αρχιτεκτονικών συστημάτων και ταυτόχρονα παράγονται εικονικά ECU για επικύρωση. Το TargetLink δημιουργεί ένα μοντέλο της λειτουργικής συμπεριφοράς και κώδικα παραγωγής σύμφωνα πάντα με τις προδιαγραφές του AUTOSAR. Στη συνέχεια η πλατφόρμα VEOS χρησιμοποιείται στον υπολογιστή κατά κύριο λόγο για επικύρωση λογισμικού και λειτουργιών των εικονικών ECU. Το Bus Implementation Tools και το Bus Manager χρησιμοποιούνται κυρίως σαν εργαλεία διαμόρφωσης για την προσομοίωση του δίαυλου επικοινωνίας (bus simulation). Σε όλες τις φάσεις η dSPACE προσφέρει ενημερωμένη βοήθεια για τις τελευταίες κυκλοφορίες του AUTOSAR καθώς και ευελιξία στην παράλληλη χρήση διαφόρων εργαλείων της αλλά και εργαλείων από άλλους κατασκευαστές.

3 Μελέτη περίπτωσης

Αρχικά, επιλέχθηκε σαν εργαλείο για την πρακτική μελέτη το πακέτο εργαλείων της Mathworks καθώς είναι ένα εργαλείο ευρέως γνωστό και εύχρηστο σε πάρα πολλούς τομείς. Το συγκεκριμένο πακέτο εργαλείων δεν περιέχει Autosar authoring tool. Επιπλέον αξίζει να σημειωθούν εμπειριές και αρκετοί περιορισμοί οι οποίοι ως επί το πλείστον είναι οι διαθέσιμοι πόροι και το κόστος για την περαιτέρω ανάπτυξη. Επειδή η ανάπτυξη του προτύπου απευθύνεται στη βιομηχανία, στους κατασκευαστές οχημάτων και εξαρτημάτων απευθείας, οι πληροφορίες που υπάρχουν διαθέσιμες, για την ανάπτυξη Autosar εφαρμογών είναι περιορισμένες και το κόστος των εργαλείων είναι ανάλογο των πελατών του.

Σε συνέχεια των παραπάνω και μελετώντας αρκετό υλικό στην βιβλιοθήκη της Mathworks, επιλέχθηκε το 'AUTOSARDemoMdl' το οποίο είναι το μοναδικό ελεύθερο μοντέλο του προτύπου το οποίο δημιουργήθηκε, από την mathworks, με το Simulink και το Embedded Coder το οποίο υπολογίζει την ταχύτητα του οχήματος και παράγει τα ανάλογα αποτελέσματα που χρειάζονται από την ECU. Τα υπόλοιπα μοντέλα-παραδείγματα που υπάρχουν διαθέσιμα είναι βασισμένα στην θεωρία και στους ορισμούς του προτύπου. Επιπλέον καθορίζει και τη λειτουργία του οδομέτρου του οχήματος. Αυτό το μοντέλο περιέχει αλγόριθμο κατάλληλο για να

χρησιμοποιηθεί σε ηλεκτρονική μονάδα οχήματος και μπορεί να παραχθεί και σε άλλο πρότυπο για ενσωματωμένα συστήματα προγενέστερο. Στο παραπάνω μοντέλο παράγουμε Autosar λογισμικό και εξαρτήματα λογισμικού έτοιμα για εφαρμογή.

3.1. Περιγραφή

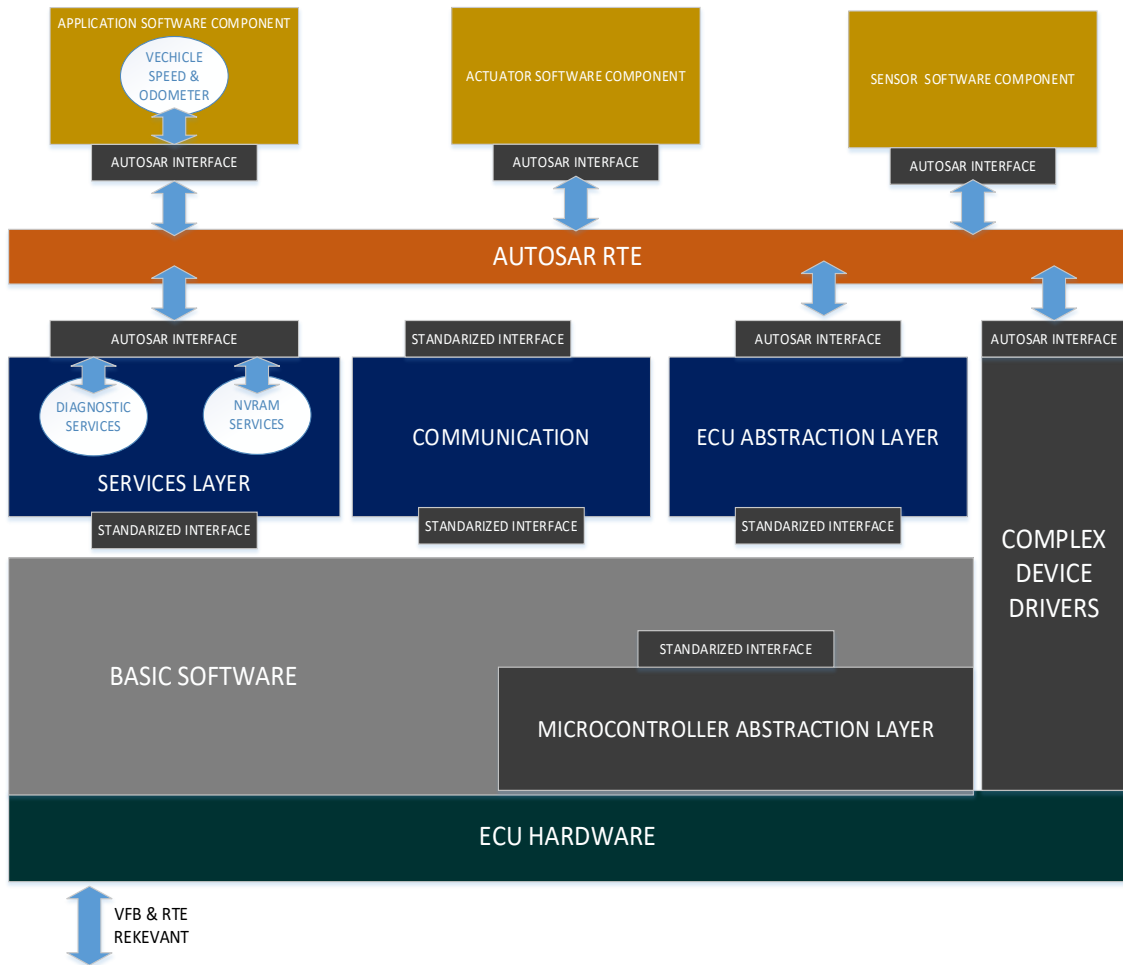
Αρχικά ας κάνουμε μια περίληψη της διαδικασίας που χρησιμοποιείται για να σχεδιαστεί και να παραχθεί το τελικό αποτέλεσμα. Η ομάδα της Mathworks χρησιμοποιώντας το Embedded coder και το Simulink model στο οποίο εισάγεται το σχέδιο που έχει σχεδιαστεί σε κάποιο Autosar authoring tool, το οποίο δεν περιέχει το πακέτο της Matlab, παράγεται κώδικας συμβατός με το Autosar πρότυπο και εξάγεται σε αρχml αρχείο.

Εμπεριέχει τα εξαρτήματα λογισμικού Autosar και πιο συγκεκριμένα τις διεργασίες του συγκεκριμένου μοντέλου που είναι υπεύθυνες για την καταγραφή της ταχύτητας του οχήματος την λειτουργία του οδομέτρου και των υπηρεσιών της μνήμης και της διάγνωσης. Οι διεργασίες αυτές είναι ουσιαστικά υποσυστήματα του λογισμικού.

Η διαδικασία παραγωγής ξεκινάει από ένα κενό μοντέλο στο Simulink model σχεδιάζοντας το διάγραμμα του λογισμικού χρησιμοποιώντας την βιβλιοθήκη των μπλοκ που προϋπάρχει στο εργαλείο της Matlab. Στη συνέχεια διαμορφώνονται οι παράμετροι των μπλοκ και οι λειτουργίες της Matlab σύμφωνα με το πρότυπο Autosar (autosar.tlc, XML schema version 4.0) του εργαλείου και σε γλώσσα προγραμματισμού C και γίνονται οι τελικές ρυθμίσεις και διαμορφώσεις των παραμέτρων του μοντέλου σαν εξάρτημα λογισμικού Autosar. Σε τελικό επίπεδο παράγεται το μοντέλο του λογισμικού αρχml, εκτελέσιμο αρχείο σε γλώσσα προγραμματισμού C.

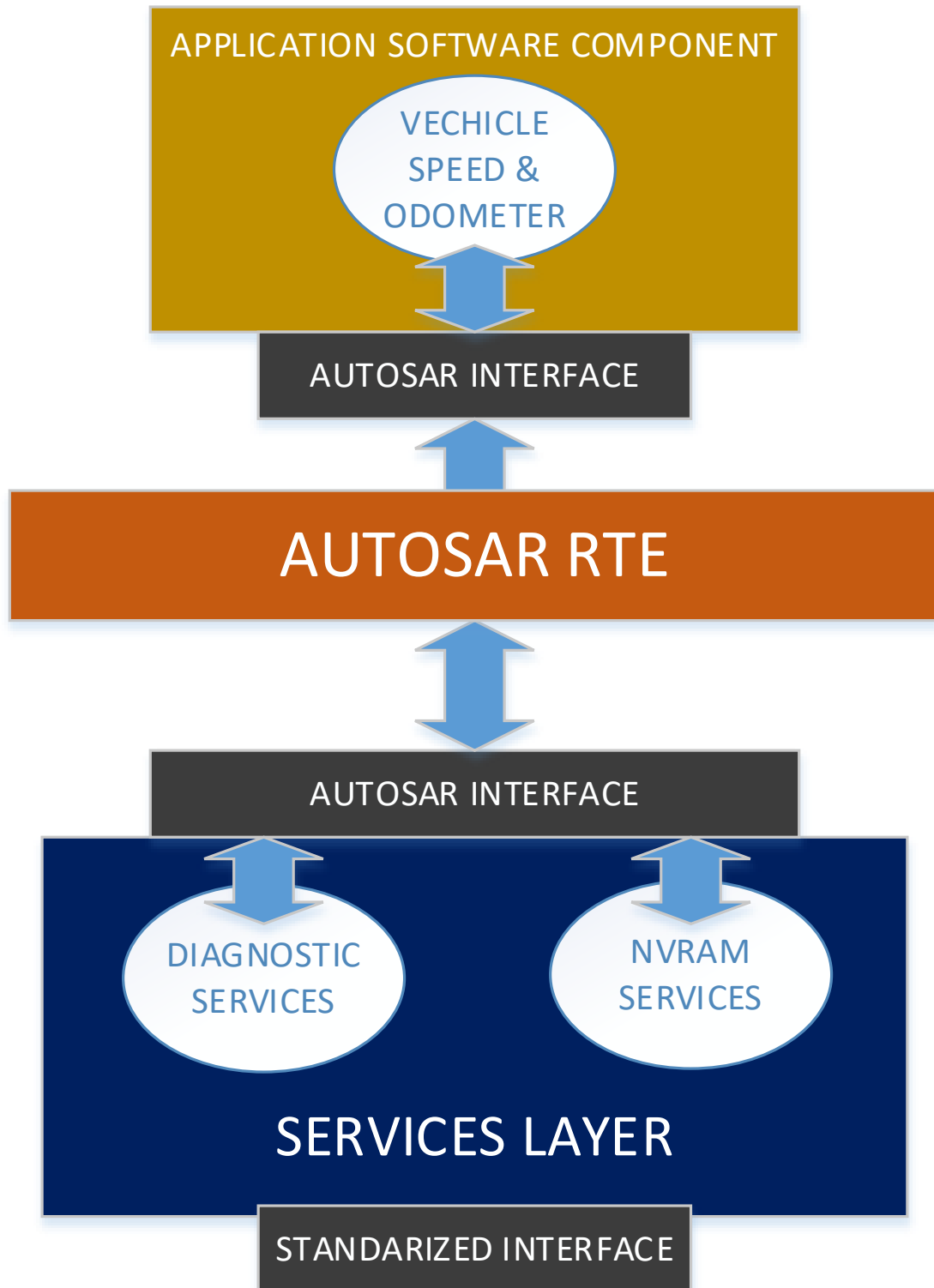
Η μεθοδολογία που ακολουθείται, πληροί τις βασικές προϋποθέσεις του προτύπου για το συγκεκριμένο μοντέλο όπου παρέχει τους τύπους των δεδομένων, τις πύλες του και τις διεπαφές του. Όπως αναλύθηκε στο βιβλιογραφικό μέρος η μεθοδολογία του προτύπου είναι αρκετά ευέλικτη συνεπώς τηρουμένων των αναλογιών δεν υπάρχει αρκετό υλικό για ανάλυση στο συγκεκριμένο μοντέλο. Είναι μια απλή διαδικασία, ρύθμιση – παραμετροποίηση και στο επόμενο βήμα παράγεται. Παρακάτω παρουσιάζονται αναλυτικά, όπως και η χαρτογράφηση του λογισμικού εξαρτήματος που παράγεται τελικά.

3.2 Αρχιτεκτονική του μοντέλου



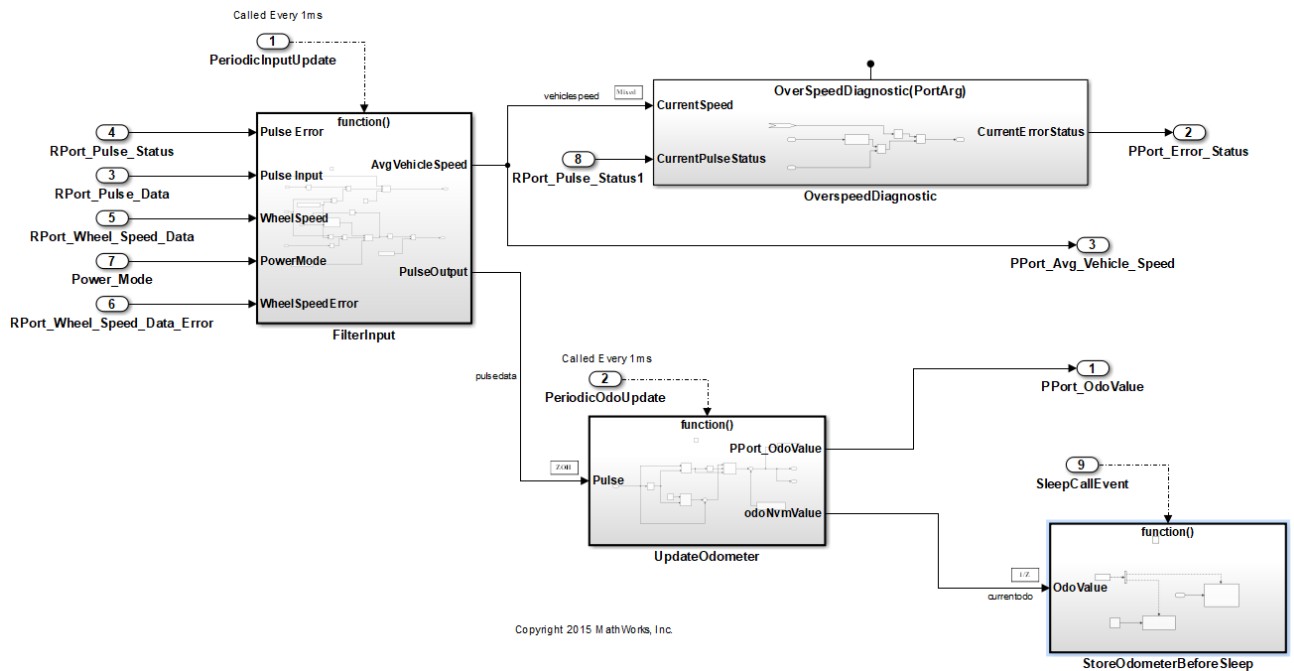
Εικόνα 11 Αρχιτεκτονική του μοντέλου εφαρμοσμένη στο Autosar

Η αρχιτεκτονική του προτύπου προσαρμοσμένη στο μοντέλο 'AUTOSARDemoMdl' απεικονίζεται στις εικόνες 11 και 11^A, ενώ στην εικόνα 12 απεικονίζεται η αρχιτεκτονική του μοντέλου όπως δημιουργείται και επεξεργάζεται στο Simulink model. Παρατηρώντας την αρχιτεκτονική προσαρμοσμένη στο συγκεκριμένο μοντέλο προκύπτει έμπρακτα ότι εφαρμόζοντας το πρότυπο με τα εργαλεία της Matlab, μετά και την παραμετροποίηση του βασικού λογισμικού και του περιβάλλοντος, λειτουργίας τα εξαρτήματα λογισμικού που παράγονται VEHICLE SPEED & ODOMETER είναι έτοιμα εξαρτήματα λογισμικού Autosar σε επίπεδο εφαρμογής και σε πλήρως ευέλικτη μορφή για περαιτέρω ανάπτυξη ή προσαρμογή και συνεργασία με άλλες εφαρμογές και λογισμικά εξαρτήματα. Παράλληλα στο επίπεδο του βασικού λογισμικού, σύμφωνα πάντα με την αρχιτεκτονική του προτύπου, δημιουργούνται διαγνωστικές υπηρεσίες που περιλαμβάνει λειτουργίες για την απαίτηση δεδομένων από τα εξαρτήματα λογισμικού και υπηρεσίες NVRam μνήμης (λειτουργία εγγραφής και ορισμού κατάστασης των μπλοκ της μνήμης). Κάθε διεπαφή αποστολέα-παραλήπτη, διεπαφή παραλήπτη-αποστολέα και διεπαφή διακόπτη λειτουργίας είναι (API) διεπαφές προγραμματισμού εφαρμογών του Autosar, δηλαδή παράγεται αυτόματα από το πρότυπο. Τα API του Autosar περιέχονται στο περιβάλλον λειτουργίας του Autosar (RTE) και παράγονται αυτόματα από την υπόλοιπη διαδικασία προγραμματισμού.



Εικόνα 11^A Αρχιτεκτονική Μοντέλου AutosarDemoMdl

3.3 Περιγραφή - Τεχνικά χαρακτηριστικά του μοντέλου



Εικόνα 12¹⁵ Simulink Map

Στην παραπάνω εικόνα μπορεί να παρατηρηθεί η αρχική χαρτογράφηση του λογισμικού που έχει δημιουργηθεί με Simulink και Embedded coder ακολουθώντας τις προϋποθέσεις μεθοδολογίας του προτύπου. Μελετώντας το λογισμικό και στο Simulink αλλά και στην παραμετροποίηση του λογισμικού σαν εξάρτημα λογισμικού του προτύπου εμπεριέχονται τα εξαρτήματά διεπαφής Autosar τα οποία παρουσιάζονται και αναλύονται παρακάτω με γνώμονα φυσικά το πρότυπο.

Autosar Data types

Μελετώντας το μοντέλο και τον τρόπο που ρυθμίζεται και παράγεται στην εργαλειοθήκη του Matlab διαπιστώνεται ότι κάθε μεταβλητή που υπάρχει στο μοντέλο, είναι τα δεδομένα του προτύπου (Autosar Data types) π.χ. η ταχύτητα των τροχών που λαμβάνεται από την αντίστοιχη πύλη δεδομένων, τα οποία όμως παράγονται στο τελικό στάδιο και περιέχονται αυτόματα σαν δεδομένα του προτύπου στο τελικό αρχείο. Πολύ απλά κάθε Autosar δεδομένο που χρησιμοποιείται στο τελικό αρχείο αντιστοιχεί σε μια μεταβλητή στο στάδιο παραγωγής κώδικα.

Πύλες εισαγωγής δεδομένων – Imports.

Οι πύλες εισαγωγής δεδομένων δημιουργούν μια πύλη δεδομένων Autosar χαρτογραφώντας τα δεδομένα εισόδου που έχουν δημιουργηθεί με βάση τη βιβλιοθήκη της Simulink. Σύμφωνα με τη θεωρία του προτύπου οι πύλες εισαγωγής απαιτούν μια διεπαφή επικοινωνίας Client-Server.

Για το συγκεκριμένο μοντέλο χρησιμοποιήθηκαν τα εξής:

- Power_Mode, με επιλογή ModeReceive τιμή, από την βιβλιοθήκη του Autosar για τις πύλες των δεδομένων. Η διεργασία της πύλης είναι να απαριθμεί τις σταθερές της συνάρτησης.
- RPort_Pulse_Data, με επιλογή ImplicitReceive τιμή, από την βιβλιοθήκη του Autosar για τις πύλες των δεδομένων. Η πύλη αυτή είναι ακέραιος αριθμός 8bit και πρακτικά είναι η πύλη που δέχεται την ταχύτητα των τροχών του οχήματος.

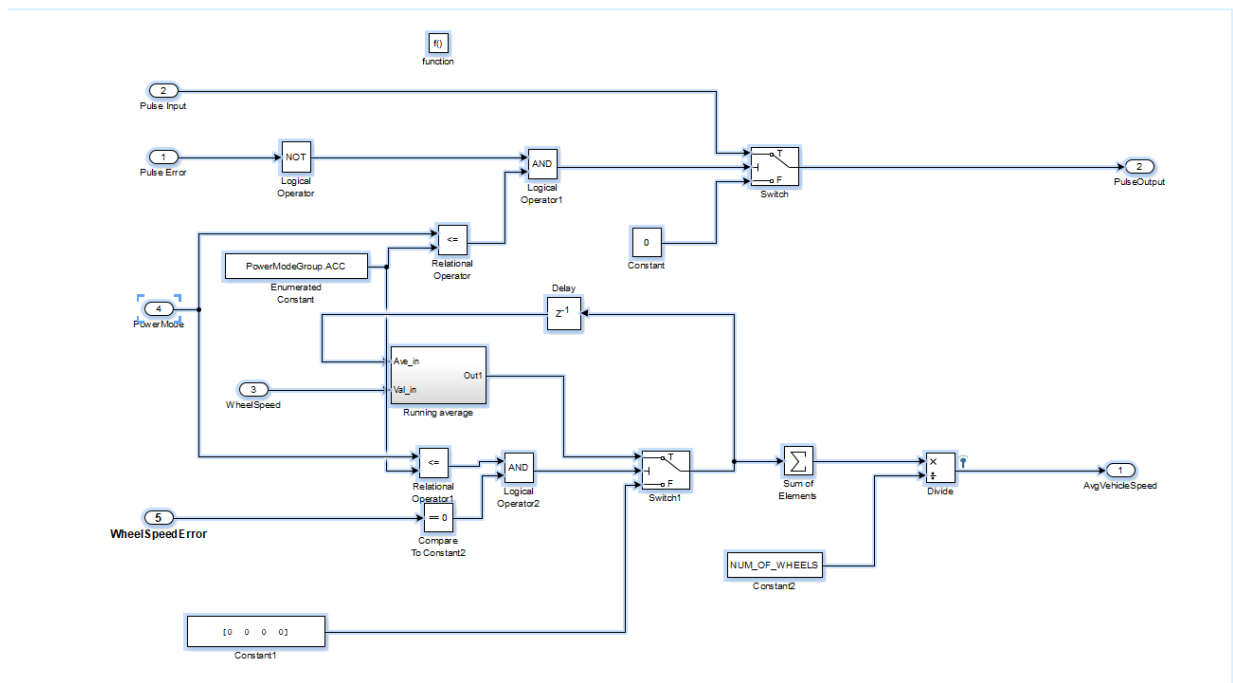
¹⁵ <https://www.mathworks.com/matlabcentral/fileexchange/52099-demo-on-autosar-code-generation-from-embedded-coder?focused=6802658&tab=example>

- RPort_Pulse_Status, με επιλογή ImplicitReceive τιμή, από την βιβλιοθήκη του Autosar για τις πύλες των δεδομένων. Η πύλη αυτή είναι μια λογική μεταβλητή που αξιολογεί τον παλμό εισόδου.
- Rport_Wheel_Speed_Data, με επιλογή ImplicitReceive τιμή, από την βιβλιοθήκη του Autosar για τις πύλες των δεδομένων. Η πύλη αυτή είναι ακέραιος αριθμός 16bit και πρακτικά είναι η πύλη που ξεκινάει η αξιολόγηση της ταχύτητας τροχών του οχήματος.
- Rport_Wheel_Speed_Data_Error, με επιλογή ErrorStatus τιμή, από την βιβλιοθήκη του Autosar για τις πύλες των δεδομένων. Η πύλη αυτή είναι ακέραιος αριθμός 8bit που οδηγεί το δεδομένο της ταχύτητας του οχήματος με σε μια λογική μεταβλητή για την αξιολόγηση του.

Πύλες εξαγωγής δεδομένων – Outports.

Οι πύλες εξαγωγής εμπεριέχουν μια διεπαφή επικοινωνίας Sender-Receiver. Για το συγκεκριμένο μοντέλο χρησιμοποιήθηκαν τα εξής:

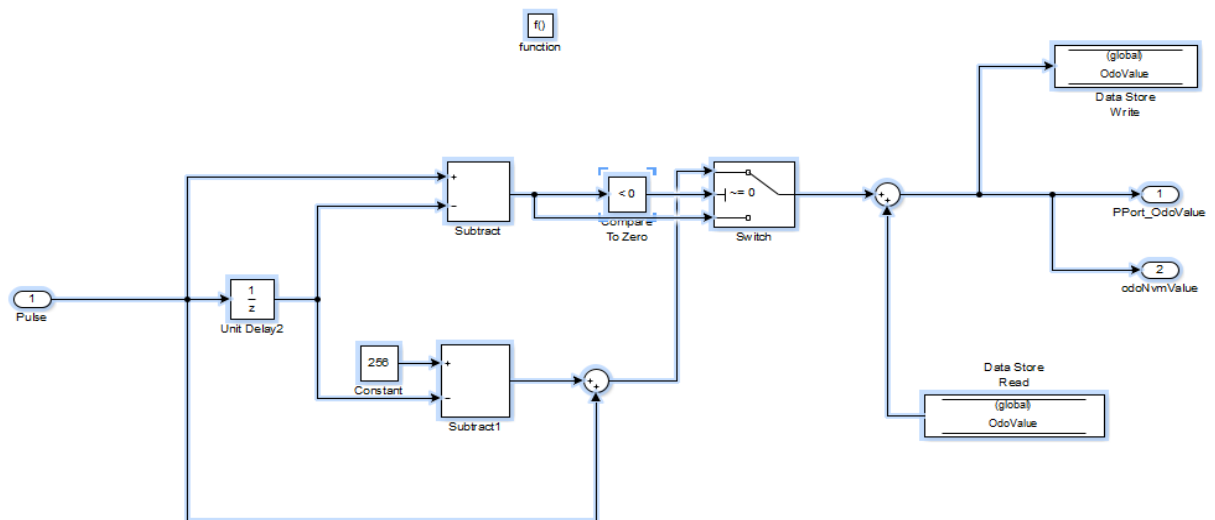
- PPort_Avg_Vehicle_Speed με επιλογή ImplicitSend τιμή, από την βιβλιοθήκη του Autosar για τις πύλες των δεδομένων. Η πύλη αυτή είναι ακέραιος αριθμός 16bit, η μέση ταχύτητα του οχήματος.
- PPort_Error_Status με επιλογή ImplicitSend τιμή, από την βιβλιοθήκη του Autosar για τις πύλες των δεδομένων. Η πύλη αυτή είναι μια λογική μεταβλητή που πρακτικά προειδοποιεί για υπερβολική ταχύτητα.
- PPort_OdoValue με επιλογή ImplicitSend τιμή, από την βιβλιοθήκη του Autosar για τις πύλες των δεδομένων. Η πύλη αυτή είναι ακέραιος αριθμός 16bit, η τιμή που θα δείχνει το ταχύμετρο του οχήματος ή η τιμή που χρησιμοποιεί η ECU για άλλες διεργασίες.
- Autosar Runnables
- Funcions – Συναρτήσεις που ορίζονται ως Autosar διεργασίες καθώς ρυθμίζονται σύμφωνα με το πρότυπο.



Εικόνα 13¹⁶ PeriodicInputUpdate Function map

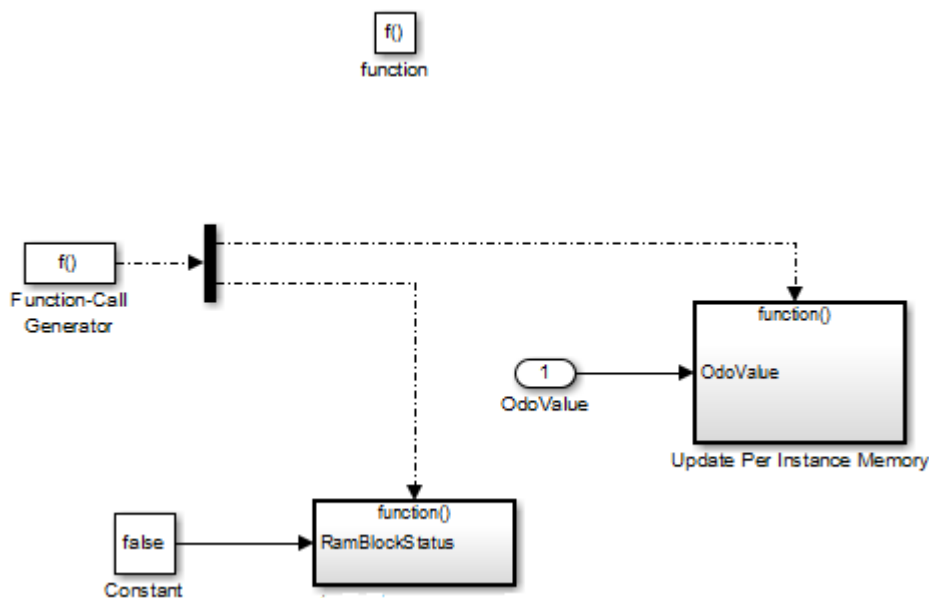
Επιπρόσθετα στο μοντέλο έχουμε:

- Exported Function: PeriodicInputUpdate καλείται κάθε 1ms, χρησιμοποιεί όλα τα δεδομένα εισόδου που περιγράψαμε παραπάνω τα επεξεργάζεται και αποδίδει τη μέση ταχύτητα του οχήματος και τον παλμό που χρησιμοποιεί σαν δεδομένο εισόδου η συνάρτηση του οδομέτρου.
- Exported Function: PeriodicOdoUpdate καλείται κάθε 1ms, χρησιμοποιεί σαν δεδομένο εισόδου το αποτέλεσμα του παλμού της προηγούμενης συνάρτησης τα επεξεργάζεται και αποδίδει την πραγματική τιμή του οδομέτρου.



Εικόνα 14¹⁶ PeriodicOdoUpdate Function map

- Exported Function: SleepCallEvent καλείται όταν η προηγούμενη συνάρτηση αποδώσει τιμή οδομέτρου και λαμβάνοντας αυτήν την τιμή φροντίζει να αποθηκευτεί στη μνήμη.



Εικόνα 15¹⁶ SleepCallEvent Function map

Data Transfers: είναι μεταβλητές εντός των διεργασιών.

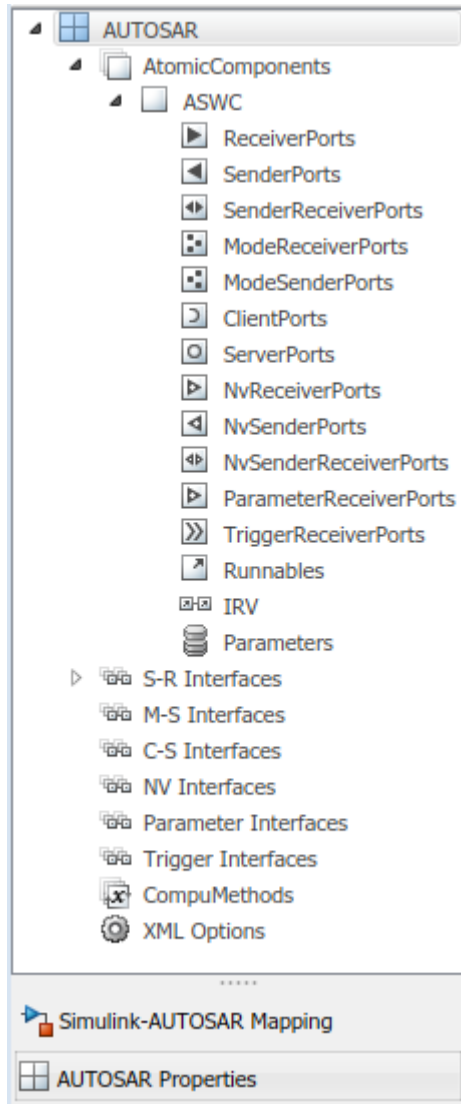
¹⁶ <https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/52099/versions/13/download/zip>

Currentodo: το οποίο εμπεριέχει την πραγματική τιμή του οδομέτρου που είναι αποθηκευμένη.

Pulsedata: το οποίο εμπεριέχει τα στοιχεία για την ταχύτητα των τροχών του οχήματος.

Vehiclespeed: το οποίο εμπεριέχει την μέση ταχύτητα του οχήματος.

Παρακάτω αναφέρονται επιγραμματικά αυτά τα εξαρτήματα όπως παρουσιάζονται με βάση την παραμετροποίηση του πρότυπου στο Simulink.



Εικόνα 16¹⁷ Ιδιότητες λογισμικού με το πρότυπο Autosar

Στο συγκεκριμένο μοντέλο συναντάται ο ατομικός τύπος εξαρτήματος λογισμικού (Atomic Software component) που είναι το αρχείο της εφαρμογής εφαρμογή: Receiver Ports (Rport_Pulse_Data, Rport_Pulse_Status, WheelSpeedPort), Sender Ports (PPort_DE5, PPort_OdoValue, VehicleSpeedPort), Mode receiver Ports (PowerModePort), Client Ports (NvmService), Server Ports (Diagnostic Port, NvmPort), Runnables (FilterInputRunnable, OverSpeedDiagnostic, Runnable_Int, StoreOdometerbeforeSleep, UpdateOdometerRunnable), IRV (currentodo, pulsedata, vehiclespeed). Παρακάτω αναφέρονται οι διεπαφές του συστήματος:

- διεπαφή αποστολέα-παραλήπτη (S-R Interface):

¹⁷ <https://www.mathworks.com/help/ecoder/examples/autosar-code-generation.html>

OdoValue, PPort_Status, PPort_Pulse_Data, PPort_Pulse_Status, PPortWheelSpeedInterface, PPortVehicleSpeed

- διεπαφή διακόπτη λειτουργίας (M-S Interface):

PowerModelInterface

- διεπαφή πελάτη εξυπηρετητή (C-S Interface):

DiagnosticInterface, NvmInterface, Nvm

Παρατηρώντας τις ιδιότητες και το αρχικό μοντέλο παρατηρείται ότι με την δημιουργία και τη διαχείριση του μοντέλου σαν πρότυπο Autosar, αυτόματα δημιουργούνται και αλλά εξαρτήματα λογισμικού που είναι πιθανό να χρειαστούν εξελίσσοντας παραπάνω την εφαρμογή. Παραδειγματικά δημιουργήθηκαν πύλες εξυπηρετητή και διαγνωστικών ελέγχων της εφαρμογής. Κατά την εξέλιξη της σε ένα ευρύτερο σύστημα του οχήματος θα μπορεί να επικοινωνεί και να συνεργάζεται με αυτό ακόμα και αν είναι ανεπτυγμένο σε άλλη πλατφόρμα - εργαλείο είτε παλαιότερο σύστημα που είναι άπλα προσαρμοσμένο στο πρότυπο.

Στο τελικό στάδιο όπου παράγεται το τελικό αρχείο Autosar σε .arxml υπάρχουν και τα υπόλοιπα αρχεία που αποτελούν το σύνολο του μοντέλου. Η αναφορά του μοντέλου δημιουργήθηκε μετά την παραγωγή και τη δημιουργία κώδικα. Τα αρχεία κώδικα για διαχείριση του μοντέλου που παράγονται είναι σε C τα AUTOSARDemoMdl.c, AUTOSARDemoMdl.h, OverSpeedDiagnostic.h, twtypes.h. Συμπεριλαμβάνονται και τα αρχεία που αποτελούν το περιβάλλον της εφαρμογής (RTE) τα οποία είναι τα Compiler.h, Platform_Types.h, Rte_ASWC.c, Rte_ASWC.h, Rte_Type.h και Std_Types.h. Το αρχείο διεπαφής που είναι και το αρχείο που χρειάζεται να εφαρμοστεί στο Hardware σε xml είναι το AUTOSARDemoMdl.arxml.

Συμπερασματικά και μελετώντας το παραπάνω μοντέλο, που έχει προγραμματιστεί από την ομάδα της Mathworks, έχει πρακτική εφαρμογή σε κάποια μονάδα ελέγχου ακόμα και εμβόλιμα σε εφαρμογή που έχει αναπτυχθεί σε άλλη ομάδα εργαλείων εφόσον οροθετηθούν οι ανάγκες του υλικού που θα χρησιμοποιηθούν και η παραμετροποίηση-ρύθμιση της εφαρμογής. Συνεπώς θα είχαμε διαθέσιμο ένα εξάρτημα λογισμικού που αναλαμβάνει τις διεργασίες του που είναι υπεύθυνες για την καταγραφή της ταχύτητας του οχήματος την λειτουργία της ενημέρωσης του οδομέτρου και των υπηρεσιών επικοινωνίας με τη μνήμη και τη διάγνωση. Η δυνατότητα ελέγχου αυτής της υπόθεσης από την μελέτη μας είναι πρακτικώς αδύνατη λόγω περιορισμένης πρόσβασης στα εργαλεία.

4. Συμπεράσματα - Περίληψη

Ανακεφαλαιώνοντας μελετώντας το θεωρητικό υπόβαθρο του προτύπου Autosar αλλά και αρκετά παραδείγματα μέσω της βιβλιοθήκης της Mathworks απαντήθηκαν οι βασικές ερωτήσεις ως τι είναι, γιατί δημιουργήθηκε, που στοχεύει και πως πρακτικά υλοποιεί τους στόχους του το πρότυπο. Αρχικά περιγράφηκε η εξέλιξη των προγενέστερων συστημάτων του Autosar έτσι ώστε να γίνει κατανοητό από πού γεννήθηκε η ιδέα του Autosar και φυσικά τι σχέση έχει αυτό με τους προκατόχους του.

Στη συνέχεια παρουσιάστηκε και αναλύθηκε το πρότυπο σε βιβλιογραφικό επίπεδο. Σε αυτό το επίπεδο μελετήθηκε η πλήρης αρχιτεκτονική σχεδίαση του λογισμικού από το επίπεδο του υλικού έως το ανώτερο επίπεδο λογισμικού, τα τεχνικά χαρακτηριστικά που υπαγορεύουν και προαπαιτούν τα όρια και τα στοιχεία για την ανάπτυξη των εφαρμογών, του στόχους και τους περιορισμούς. Επιπλέον μελετήθηκε η μεθοδολογία και τα εργαλεία ανάπτυξης και εφαρμογής από το στάδιο του προγραμματισμού υλικού έως και την προσομοίωση.

Σε τελικό επίπεδο μελετήθηκε η πρακτική εφαρμογή του προτύπου κάτι το οποίο έγινε μελετώντας το μοντέλο 'AUTOSARDemoMdl' της βιβλιοθήκης της Mathworks το οποίο είναι αναπτυγμένο με το toolbox του MATLAB, το οποίο και περιέχει τα δυο τρίτα των εργαλείων που απαιτούνται για την ανάπτυξη του προτύπου. Παρουσιάζεται, αναλύεται και σχολιάζεται σε πρακτικό επίπεδο η ανάπτυξη του μοντέλου, καθώς και η εφαρμογή του σε σχέση με την αρχιτεκτονική του μοντέλου, τα τεχνικά χαρακτηριστικά που χρησιμοποιούνται σύμφωνα πάντα με όσα μελετήθηκαν σε θεωρητικό επίπεδο.

Αρχικά η ανεξαρτησία του προτύπου σε σχέση με το υλικό είναι πραγματικά δυνατό να επιτευχθεί. Οι απαιτήσεις του προτύπου είναι όσο το δυνατόν ελάχιστες και πολύ ευέλικτες δίνοντας έτσι τη δυνατότητα να προσαρμόζεται εκείνο στο υλικό και όχι το αντίθετο χωρίς να ξεχνάει την ύπαρξη προγενέστερης τεχνολογίας παρέχοντας τη δυνατότητα να προσαρμοστεί εμβόλιμα στο πρότυπο. Επιπλέον, άλλος ένας επιτυχημένος στόχος, με επιφύλαξη μεν λόγω της εργαλειοθήκης που αναλύεται παραπάνω, είναι η ευκολία μεταφοράς και επεξεργασίας των εξαρτημάτων λογισμικού του προτύπου κάτι το οποίο το οφείλει στην αρχιτεκτονική, η οποία ευνοεί τον μετασχηματισμό του λογισμικού και των περιεχομένων του.

Σε συνέχεια του προηγούμενου σχολίου η εργαλειοθήκη του προτύπου περιλαμβάνει αρκετά εργαλεία τα οποία όσο περνάει ο καιρός αυξάνονται και εμπλουτίζονται σε όλα τα επίπεδα του προτύπου από την σχεδίαση μέχρι την προσομοίωση του αλλά σαν εργαλεία δεν ακολουθούν κάποιο πρότυπο. Αυτό έχει αποτέλεσμα να είναι ανεξάρτητα και να γεννούν πολλά ερωτήματα ώστε κατά πόσο μια πολύπλοκη εφαρμογή που παράχθηκε με μια εργαλειοθήκη θα είναι απόλυτα διαχειρίσιμη και εύκολο να μεταποιηθεί από άλλη εργαλειοθήκη. Αντίστοιχα υπάρχουν ερωτηματικά στη διαχείριση, ενημέρωση και μεταποίηση όταν υπάρχουν διαφορετικές εκδόσεις του προτύπου. Επιπλέον παρατηρήθηκε ότι σε επίπεδο λειτουργικού συστήματος παρόλο που υπάρχει ο απαραίτητος σχεδιασμός δεν υπάρχει εξέλιξη και εφαρμογή στην χαρτογράφηση των εξαρτημάτων του λογισμικού.

Η επόμενη μέρα περιλαμβάνει εξέλιξη, ήδη υπάρχουν αρκετές εκδόσεις του προτύπου από την μέρα που ξεκίνησε ανταγωνισμός, έχει ήδη ανακοινωθεί η εξέλιξη πλατφόρμας που θα επιτρέπει τη συνεργασία του προτύπου με τον ανταγωνισμό. Ανταγωνισμός υπάρχει παρόλο που όλοι σχεδόν οι κατασκευαστές οχημάτων και εργαλείων, άμεσα ή έμμεσα συμμετέχουν στο πρότυπο. Για παράδειγμα πρόσφατα παρουσιάστηκε το Waymo, προϊόν της google το οποίο ουσιαστικά, προσωπική άποψη, είναι ανάπτυξη προτύπου λογισμικού για οχήματα. Ακόμα, παρατηρώντας την Tesla και τον τρόπο που εξελίσσεται γιατί στο μέλλον το ανταγωνιστικό της πλεονέκτημα να μην είναι το λογισμικό της;

Προσωπική άποψη και εκτίμηση είναι ότι υπάρχει έλλειψη εξέλιξης στο ανώτερο επίπεδο του λογισμικού παρόλο που υπάρχουν όλες οι προϋποθέσεις. Δηλαδή για να γίνει ακόμα πιο εύχρηστο το πρότυπο, καλό είναι να δημιουργηθεί ένα πρότυπο λειτουργικό σύστημα στο επίπεδο εφαρμογής με γραφικό περιβάλλον λειτουργίας, όπου εκεί θα υπάρχουν και τα εργαλεία και θα επιτρέπει σε όλους τους ενδιαφερόμενους να προγραμματίζουν εύκολα στο πρότυπο, προσεγγίζοντας έτσι περισσότερους προγραμματιστές. Για παράδειγμα, συγκρίνοντας το με την εξέλιξη των προσωπικών υπολογιστών το λειτουργικό σύστημα και τα εργαλεία είναι επιπέδου kernel ενώ αυτό που χρειάζεται είναι Windows / Linux με πρόσβαση από κάθε ενδιαφερόμενο για να απογειωθεί η εξέλιξη αυτών των συστημάτων.

Σε τελικό επίπεδο ο στόχος, που εκτιμάται ότι πρέπει να θεσπιστεί, είναι το πρότυπο να παρέχει πλήρες περιβάλλον λειτουργίας ενός οχήματος προς ανάπτυξη, πάντα ανοικτού κώδικα έτσι ώστε να μην ανταγωνίζεται τα εργαλεία που έχουν επενδύσει σε αυτό αλλά να το βελτιώνουν και να εξειδικεύονται σε ποιο ειδικές ανάγκες. Η μη ανάπτυξη εργαλειοθήκης-λειτουργικού περιβάλλοντος μέχρι τώρα μπορεί να γίνει το καταστροφικό μειονέκτημα του εάν βρεθεί μια πιο έτοιμη λύση. Για παράδειγμα, ένας κατασκευαστής που θέλει να σχεδιάσει ένα όχημα να οριοθετεί το υλικό (μονάδες ελέγχου, αισθητήρες ενεργοποιητές κτλ.) και στη συνέχεια να τα ενώνει όλα και να τα διαχειρίζεται από μια πλατφόρμα σαν σημείο εκκίνησης. Τώρα το πρότυπο, έλλειψη του σημείου εκκίνησης, απλά εξασφαλίζει την ομαλή συνεργασία των μερών του. Τη σημερινή εποχή που στόχος είναι τα οχήματα να κινούνται αυτόνομα, η πολυπλοκότητα αυξάνεται με γεωμετρική πρόοδο κάτι που είναι πολύ διαφορετικό σε σχέση με τα προβλήματα που υπήρχαν όταν δημιουργήθηκε.

5. Βιβλιογραφία

- [1] Devendra Rai, T. K. Jestin, Lev Vitkin, (2008) *Model-Based Development of AUTOSAR Compliant, Applications: Exterior Lights, Module Case Study*, SAE publication
- [2] Hansjörg Lipowsky, Emin Arpaci (2007), *Copper in the Automotive Industry*, WILEY-VCH Verlag GmbH & Co. KGaA
- [3] Jürgen Valldorf · Wolfgang Gessner, (2006), *Advanced Microsystems for Automotive Applications*, Springer-Verlag Berlin Heidelberg
- [4] Dr. Joachim Schlosser, (2008), *Development of AUTOSAR Software Components with Model-Based Design*, Application Engineering The MathWorks GmbH
- [5] Guido Sandmann, Richard Thompson (2008), *Development of AUTOSAR Software Components within Model-Based Design*, The MathWorks, Inc
- [6] Nicolas Navet - Francoise Simonot-Lion (2009), *Automotive Embedded Systems Handbook*, CRC Press by Taylor & Francis Group, LLC
- [7] Kassakian, J.G; Wolf, H.-C.; Miller, J.M., Hurton, C.J. (1996), *Automotive electrical systems circa 2005* – IEEE Spectrum
- [8] Thomas Bachmann (7/30/2009), *FIBEX XML format and AUTOSAR development*, Διαθέσιμο στην ιστοσελίδα eetimes.com: https://www.eetimes.com/document.asp?doc_id=1276593 [10/08/2016]
- [9] Victor Reyes, (2013), Evolution of automotive software testing, Διαθέσιμο στην ιστοσελίδα του embedded.com:
- [10] <http://www.embedded.com/electronics-blogs/virtual-prototyping-ales/4416104/Automotive-software-testing-evolution> [4/3/2017]
- [11] An introduction to AUTOSAR, Διαθέσιμο στην ιστοσελίδα του The ReTiS Lab:
- [12] https://retis.sssup.it/sites/default/files/lesson19_autosar.pdf [4/3/2017]
- [13] Balaji Kulkarni, (2017), AUTOSAR – Automotive Open Systems Architecture, Διαθέσιμο στην ιστοσελίδα του www.engineersgarage.com:
- [14] <https://www.engineersgarage.com/articles/autosar-automotive-open-systems-architecture> [10/08/2017]
- [15] Autosar Technical Overview, Διαθέσιμο στην ιστοσελίδα του autosar.org: <http://www.autosar.org/about/technical-overview/> [05/08/2016]
- [16] AUTOSAR solution, Διαθέσιμο στην ιστοσελίδα του vector.com: https://vector.com/vi_autosar_solutions_en.html [15/09/2017]
- [17] AUTOSAR, Διαθέσιμο στην ιστοσελίδα του mathworks.com: <https://www.mathworks.com/solutions/automotive/standards/autosar.html>
- [18] AUTOSAR products, Διαθέσιμο στην ιστοσελίδα του arccore.com: <https://www.arccore.com/products> [15/09/2017]
- [19] ECU Design with AUTOSAR, Διαθέσιμο στην ιστοσελίδα του mentor.com: <https://www.mentor.com/embedded-software/automotive/autosar> [15/09/2017]
- [20] AUTOSAR solution, Διαθέσιμο στην ιστοσελίδα του etas.com: https://www.etas.com/en/products/applications_autosar.php [20/09/2017]
- [21] AUTOSAR Technology at dSPACE, Διαθέσιμο στην ιστοσελίδα του dspace.com: https://www.dspace.com/shared/data/pdf/2017/dSPACE_AUTOSAR-Technology-2017_English.pdf [20/09/2017]
- [22] Demo on AUTOSAR Code Generation From Embedded Coder, Διαθέσιμο στην ιστοσελίδα του mathworks.com: <https://www.mathworks.com/matlabcentral/fileexchange/52099-demo-on-autosar-code-generation-from-embedded-coder?focused=6802658&tab=example> [22/10/2017]
- [23] Demo on AUTOSAR Code Generation From Embedded Coder, Διαθέσιμο στην ιστοσελίδα του mathworks.com: <https://www.mathworks.com/matlabcentral/mlcdownloads/downloads/submissions/52099/versions/13/download/zip> [22/10/2017]
- [24] Embedded Coder Support Package for AUTOSAR Standard, Διαθέσιμο στην ιστοσελίδα του mathworks.com: [www.mathworks.comhttps://www.mathworks.com/help/supportpkg/autosarstandard/examples/model-for-autosar-platform.html](https://www.mathworks.com/help/supportpkg/autosarstandard/examples/model-for-autosar-platform.html) [29/10/2017]
- [25] Autosar, Διαθέσιμο στην ιστοσελίδα του mathworks.com: <https://www.mathworks.com/help/ecoder/autosar-software-components.html> [29/10/2017]

[26] AUTOSAR Support from Simulink and Embedded Coder, Διαθέσιμο στην ιστοσελίδα του mathworks.com:

<https://www.mathworks.com/videos/autosar-support-from-simulink-and-embedded-coder-121482.html> [29/10/2017]

[27] From Simulink to AUTOSAR Production Code, Διαθέσιμο στην ιστοσελίδα του mathworks.com:

<https://www.mathworks.com/videos/from-simulink-to-autosar-production-code-102308.html> [29/10/2017]

6. Παράρτημα

Τελική αναφορά του μοντέλου της μελέτης

Περιεχόμενα της αναφοράς

[Summary](#)

[Subsystem Report](#)

[Code Interface Report](#)

[Traceability Report](#)

[Static Code Metrics Report](#)

[Code Replacements Report](#)

Generated Code

[– **Model files**
]

[AUTOSARDemoMdl.c](#)

[AUTOSARDemoMdl.h](#)

[– **Shared files**
]

[OverSpeedDiagnostic.h](#)

[rtwtypes.h](#)

[– **Interface files**
]

[AUTOSARDemoMdl.arxml](#)

[– **RTE files**
]

[Compiler.h](#)

[Platform Types.h](#)

[Rte ASWC.c](#)

[Rte ASWC.h](#)

[Rte Type.h](#)

[Std Types.h](#)

Αναφορά του κώδικα της διεπαφής

Code Interface Report for model

Table of Contents

- Entry Point Functions
- Inports
- Outports
- Interface Parameters
- Data Stores

Entry Point FunctionsFunction: Runnable_Init

Prototype	void Runnable_Init(void)
Description	Initialization entry point of generated code
Timing	Must be called exactly once
Arguments	None
Return value	None
Header file	

Function: FilterInputRunnable

Prototype	void FilterInputRunnable(void)
Description	Exported function: <u><Root>/PeriodicInputUpdate</u>
Timing	Must be called periodically, every 0.001 seconds
Arguments	None
Return value	None
Header file	

Function: StoreOdometerBeforeSleep

Prototype	void StoreOdometerBeforeSleep(void)
Description	Exported function: <u><Root>/SleepCallEvent</u>
Timing	Can be called at any time
Arguments	None
Return value	None

Header file

Function: UpdateOdometerRunnable

Prototype **void UpdateOdometerRunnable(void)**
 Description Exported function: <Root>/PeriodicOdoUpdate
 Timing Must be called periodically, every 0.001 seconds
 Arguments None
 Return value None
 Header file

Function: OverSpeedDiagnostic

Prototype **void OverSpeedDiagnostic(uint8_T PortArg)**

pe

Description Simulink function

Timing Can be called at any time

Arguments	#	Name	Data Type	Description
	1	PortArg	uint8_T	Input Argument

Return value None

Header file OverSpeedDiagnostic.h

Inports

[-]

Block Name	Code Identifier	Data Type	Dimension
<u><Root>/RPort Pulse Data</u>	<i>Require port, interface: sender-receiver</i>	uint8_T	1
<u><Root>/RPort Pulse Status</u>	<i>Require port, interface: sender-receiver</i>	boolean_T	1

<u><Root>/RPort Wheel Speed Data</u>	<i>Require port, interface: sender-receiver</i>	real_T	[4]
<u><Root>/RPort Wheel Speed Data Error</u>	<i>ErrorStatus</i>	uint8_T	1
<u><Root>/Power Mode</u>	<i>Require port, interface: mode-switch</i>	PowerModeGroup	1
<u><Root>/RPort Pulse Status1</u>	<i>Require port, interface: sender-receiver</i>	boolean_T	1

Outputs

[-]

Block Name	Code Identifier	Data Type	Dimension
<u><Root>/PPort OdoValue</u>	<i>Provide port, interface: sender-receiver</i>	uint16_T	1
<u><Root>/PPort Error Status</u>	<i>Provide port, interface: sender-receiver</i>	boolean_T	1
<u><Root>/PPort Avg Vehicle Speed</u>	<i>Provide port, interface: sender-receiver</i>	real_T	1

Interface Parameters

No interface/tunable parameters in model.

Data Stores

Data Store Source	Code Identifier	Data Type	Dimension
OdoValue	OdoValue	uint16_T	1

Δείγμα του αρχείου .arxml του μοντέλου**.arxml**

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3 Auto generated XML Component Description
4 -->
5 <AUTOSAR xmlns="http://autosar.org/schema/r4.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://autosar.org/schema/r4.0 AUTOSAR_4-0-3.xsd">
6 <AR-PACKAGES>
7 <AR-PACKAGE UUID="a66658d5-10b3-437d-9089-f396c77dee8f">
8 <SHORT-NAME>pkg</SHORT-NAME>
9 <AR-PACKAGES>
10 <AR-PACKAGE UUID="7ee37065-4a91-5fb2-8658-173dbbc3f7cf">
11 <SHORT-NAME>swc</SHORT-NAME>
12 <ELEMENTS>
13 <APPLICATION-SW-COMPONENT-TYPE UUID="33724001-b686-4088-be58-
  08907eca205c">
14 <SHORT-NAME>ASWC</SHORT-NAME>
15 <PORTS>
16 <R-PORT-PROTOTYPE UUID="b8d97e8d-4aee-45e0-a61c-c21caf1dfaa8">
17 <SHORT-NAME>RPort_Pulse_Data</SHORT-NAME>
18 <REQUIRED-COM-SPECS>
19 <NONQUEUED-RECEIVER-COM-SPEC>
20 <DATA-ELEMENT-REF DEST="VARIABLE-DATA-
  PROTOTYPE">/pkg/if/RPort_Pulse_Data/RPort_Pulse_Data</DATA-ELEMENT-REF>
21 <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
22 <USES-END-TO-END-PROTECTION>>false</USES-END-TO-END-PROTECTION>
23 <ALIVE-TIMEOUT>60</ALIVE-TIMEOUT>
24 <ENABLE-UPDATE>>false</ENABLE-UPDATE>
25 <HANDLE-NEVER-RECEIVED>>false</HANDLE-NEVER-RECEIVED>
26 <HANDLE-TIMEOUT-TYPE>NONE</HANDLE-TIMEOUT-TYPE>
27 <INIT-VALUE>
29 <CONSTANT-REFERENCE>
30 <SHORT-LABEL>DefaultInitValue_UInt8</SHORT-LABEL>
32 <CONSTANT-REF DEST="CONSTANT-
  SPECIFICATION">/pkg/ASWC_dt/Ground/DefaultInitValue_UInt8</CONSTANT-REF>
33 </CONSTANT-REFERENCE>
34 </INIT-VALUE>
35 </NONQUEUED-RECEIVER-COM-SPEC>

```

```

36 </REQUIRED-COM-SPECS>
37 <REQUIRED-INTERFACE-TREF DEST="SENDER-RECEIVER-
INTERFACE">/pkg/if/RPort_Pulse_Data</REQUIRED-INTERFACE-TREF>
38 </R-PORT-PROTOTYPE>
39 <R-PORT-PROTOTYPE UUID="b7cdced0-bc5e-4d18-84ed-d96cca6f3a90">
40 <SHORT-NAME>RPort_Pulse_Status</SHORT-NAME>
41 <REQUIRED-COM-SPECS>
43 <NONQUEUED-RECEIVER-COM-SPEC>
44 <DATA-ELEMENT-REF DEST="VARIABLE-DATA-
PROTOTYPE">/pkg/if/RPort_Pulse_Status/RPort_Pulse_Status</DATA-ELEMENT-
REF>
45 <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
46 <USES-END-TO-END-PROTECTION>>false</USES-END-TO-END-PROTECTION>
47 <ALIVE-TIMEOUT>60</ALIVE-TIMEOUT>
48 <ENABLE-UPDATE>>false</ENABLE-UPDATE>
50 <HANDLE-NEVER-RECEIVED>>false</HANDLE-NEVER-RECEIVED>
51 <HANDLE-TIMEOUT-TYPE>NONE</HANDLE-TIMEOUT-TYPE>
52 <INIT-VALUE>
53 <CONSTANT-REFERENCE>
54 <SHORT-LABEL>DefaultInitValue_Boolean</SHORT-LABEL>
55 <CONSTANT-REF DEST="CONSTANT-
SPECIFICATION">/pkg/ASWC_dt/Ground/DefaultInitValue_Boolean</CONSTANT-
REF>
56 </CONSTANT-REFERENCE>
57 </INIT-VALUE>
58 </NONQUEUED-RECEIVER-COM-SPEC>
59 <NONQUEUED-RECEIVER-COM-SPEC>
60 <DATA-ELEMENT-REF DEST="VARIABLE-DATA-
PROTOTYPE">/pkg/if/RPort_Pulse_Status/RPort_Pulse_Status1</DATA-ELEMENT-
REF>
61 <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
62 <USES-END-TO-END-PROTECTION>>false</USES-END-TO-END-PROTECTION>
63 <ALIVE-TIMEOUT>60</ALIVE-TIMEOUT>
64 <ENABLE-UPDATE>>false</ENABLE-UPDATE>
65 <HANDLE-NEVER-RECEIVED>>false</HANDLE-NEVER-RECEIVED>
66 <HANDLE-TIMEOUT-TYPE>NONE</HANDLE-TIMEOUT-TYPE>
67 <INIT-VALUE>
68 <CONSTANT-REFERENCE>
69 <SHORT-LABEL>DefaultInitValue_Boolean</SHORT-LABEL>
70 <CONSTANT-REF DEST="CONSTANT-
SPECIFICATION">/pkg/ASWC_dt/Ground/DefaultInitValue_Boolean</CONSTANT-

```

```

REF>
71 </CONSTANT-REFERENCE>
72 </INIT-VALUE>
73 </NONQUEUED-RECEIVER-COM-SPEC>
74 </REQUIRED-COM-SPECS>
75 <REQUIRED-INTERFACE-TREF DEST="SENDER-RECEIVER-
INTERFACE">/pkg/if/RPort_Pulse_Status</REQUIRED-INTERFACE-TREF>
76 </R-PORT-PROTOTYPE>
77 <R-PORT-PROTOTYPE UUID="e9341fe6-4083-45f6-b2a3-9525f3479ad1">
78 <SHORT-NAME>WheelSpeedPort</SHORT-NAME>
79 <REQUIRED-COM-SPECS>
80 <NONQUEUED-RECEIVER-COM-SPEC>
81 <DATA-ELEMENT-REF DEST="VARIABLE-DATA-
PROTOTYPE">/pkg/if/RPortWheelSpeedInterface/WheelSpeedArray</DATA-
ELEMENT-REF>
82 <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
83 <USES-END-TO-END-PROTECTION>>false</USES-END-TO-END-PROTECTION>
84 <ALIVE-TIMEOUT>60</ALIVE-TIMEOUT>
85 <ENABLE-UPDATE>>false</ENABLE-UPDATE>
86 <HANDLE-NEVER-RECEIVED>>false</HANDLE-NEVER-RECEIVED>
87 <HANDLE-TIMEOUT-TYPE>NONE</HANDLE-TIMEOUT-TYPE>
88 <INIT-VALUE>
89 <CONSTANT-REFERENCE>
90 <SHORT-LABEL>DefaultInitValue_rt_Array_Double_4</SHORT-LABEL>
91 <CONSTANT-REF DEST="CONSTANT-
SPECIFICATION">/pkg/ASWC_dt/Ground/DefaultInitValue_rt_Array_Double_4</CONSTANT-
REF>
92 </CONSTANT-REFERENCE>
93 </INIT-VALUE>
94 </NONQUEUED-RECEIVER-COM-SPEC>
95 </REQUIRED-COM-SPECS>
96 <REQUIRED-INTERFACE-TREF DEST="SENDER-RECEIVER-
INTERFACE">/pkg/if/RPortWheelSpeedInterface</REQUIRED-INTERFACE-TREF>
97 </R-PORT-PROTOTYPE>
98 <P-PORT-PROTOTYPE UUID="aa28c973-2347-48a9-bffe-130da678c9ea">
99 <SHORT-NAME>PPort_OdoValue</SHORT-NAME>
100 <PROVIDED-COM-SPECS>
101 <NONQUEUED-SENDER-COM-SPEC>
102 <DATA-ELEMENT-REF DEST="VARIABLE-DATA-
PROTOTYPE">/pkg/if/OdoValue/OdoValue</DATA-ELEMENT-REF>

```



```
103 <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
104 <USES-END-TO-END-PROTECTION>>false</USES-END-TO-END-PROTECTION>
105 <INIT-VALUE>
106 <CONSTANT-REFERENCE>
107 <SHORT-LABEL>DefaultInitValue_UInt16</SHORT-LABEL>
108 <CONSTANT-REF DEST="CONSTANT-
SPECIFICATION">/pkg/ASWC_dt/Ground/DefaultInitValue_UInt16</CONSTANT-REF>
109 </CONSTANT-REFERENCE>
110 </INIT-VALUE>
111 </NONQUEUED-SENDER-COM-SPEC>
112 </PROVIDED-COM-SPECS>
113 <PROVIDED-INTERFACE-TREF DEST="SENDER-RECEIVER-
INTERFACE">/pkg/if/OdoValue</PROVIDED-INTERFACE-TREF>
114 </P-PORT-PROTOTYPE>
115 <P-PORT-PROTOTYPE UUID="91b2bf67-0cc6-4d79-aaef-ea5e72f97d63">
116 <SHORT-NAME>PPort_DE5</SHORT-NAME>
117 <PROVIDED-COM-SPECS>
118 <NONQUEUED-SENDER-COM-SPEC>
119 <DATA-ELEMENT-REF DEST="VARIABLE-DATA-
PROTOTYPE">/pkg/if/PPort_Status/PPort_Status</DATA-ELEMENT-REF>
120 <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
121 <USES-END-TO-END-PROTECTION>>false</USES-END-TO-END-PROTECTION>
122 <INIT-VALUE>
123 <CONSTANT-REFERENCE>
124 <SHORT-LABEL>DefaultInitValue_Boolean</SHORT-LABEL>
125 <CONSTANT-REF DEST="CONSTANT-
SPECIFICATION">/pkg/ASWC_dt/Ground/DefaultInitValue_Boolean</CONSTANT-
REF>
126 </CONSTANT-REFERENCE>
127 </INIT-VALUE>
128 </NONQUEUED-SENDER-COM-SPEC>
129 </PROVIDED-COM-SPECS>
130 <PROVIDED-INTERFACE-TREF DEST="SENDER-RECEIVER-
INTERFACE">/pkg/if/PPort_Status</PROVIDED-INTERFACE-TREF>
131 </P-PORT-PROTOTYPE>
132 <P-PORT-PROTOTYPE UUID="b4cc7fea-c888-464f-9fd6-267e29f8f4a4">
133 <SHORT-NAME>VehicleSpeedPort</SHORT-NAME>
134 <PROVIDED-COM-SPECS>
135 <NONQUEUED-SENDER-COM-SPEC>
136 <DATA-ELEMENT-REF DEST="VARIABLE-DATA-
```

```

    PROTOTYPE"/>/pkg/if/PPortVehicleSpeed/VehicleSpeedData</DATA-ELEMENT-REF>
137 <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
138 <USES-END-TO-END-PROTECTION>>false</USES-END-TO-END-PROTECTION>
139 <INIT-VALUE>
140 <CONSTANT-REFERENCE>
141 <SHORT-LABEL>DefaultInitValue_Double</SHORT-LABEL>
142 <CONSTANT-REF                                DEST="CONSTANT-
    SPECIFICATION"/>/pkg/ASWC_dt/Ground/DefaultInitValue_Double</CONSTANT-REF>
143 </CONSTANT-REFERENCE>
144 </INIT-VALUE>
145 </NONQUEUED-SENDER-COM-SPEC>
146 </PROVIDED-COM-SPECS>
147 <PROVIDED-INTERFACE-TREF                                DEST="SENDER-RECEIVER-
    INTERFACE"/>/pkg/if/PPortVehicleSpeed</PROVIDED-INTERFACE-TREF>
148 </P-PORT-PROTOTYPE>
149 <R-PORT-PROTOTYPE UUID="63a8eeca-4198-4dd2-92b9-f9a99e326342">
150 <SHORT-NAME>PowerModePort</SHORT-NAME>
151 <REQUIRED-INTERFACE-TREF                                DEST="MODE-SWITCH-
    INTERFACE"/>/Modes/PowerModeInterface</REQUIRED-INTERFACE-TREF>
152 </R-PORT-PROTOTYPE>
153 <R-PORT-PROTOTYPE UUID="f0862a11-a916-5f69-f136-40874076b2b7">
154 <SHORT-NAME>NvmService</SHORT-NAME>
155 <REQUIRED-INTERFACE-TREF                                DEST="CLIENT-SERVER-
    INTERFACE"/>/AUTOSAR/Services/NvM</REQUIRED-INTERFACE-TREF>
156 </R-PORT-PROTOTYPE>
157 <P-PORT-PROTOTYPE UUID="ac19443b-e199-489d-95f3-1c399d3b1efb">
158 <SHORT-NAME>NvmPort</SHORT-NAME>
159 <PROVIDED-INTERFACE-TREF                                DEST="CLIENT-SERVER-
    INTERFACE"/>/pkg/if/NvmInterface</PROVIDED-INTERFACE-TREF>
160 </P-PORT-PROTOTYPE>
161 <P-PORT-PROTOTYPE UUID="aefb885b-e534-43df-bfe3-0f14490e84de">
162 <SHORT-NAME>DiagnosticPort</SHORT-NAME>
163 <PROVIDED-COM-SPECS>
164 <SERVER-COM-SPEC>
165 <OPERATION-REF                                DEST="CLIENT-SERVER-
    OPERATION"/>/pkg/swc/DiagnosticInterface/DataRequest</OPERATION-REF>
166 <QUEUE-LENGTH>1</QUEUE-LENGTH>
167 </SERVER-COM-SPEC>
168 </PROVIDED-COM-SPECS>
169 <PROVIDED-INTERFACE-TREF                                DEST="CLIENT-SERVER-

```

```

INTERFACE"/>/pkg/swc/DiagnosticInterface</PROVIDED-INTERFACE-TREF>
170 </P-PORT-PROTOTYPE>
171 </PORTS>
172 <INTERNAL-BEHAVIORS>
173 <SWC-INTERNAL-BEHAVIOR UUID="118fce4b-e4e2-4995-be96-057b0b1d0254">
174 <SHORT-NAME>ASWC</SHORT-NAME>
175 <DATA-TYPE-MAPPING-REFS>
176 <DATA-TYPE-MAPPING-REF DEST="DATA-TYPE-MAPPING-
SET"/>/pkg/ASWC_dt/DataTypeMappings/ASWCDataTypeMappingsSet</DATA-TYPE-
MAPPING-REF>
177 </DATA-TYPE-MAPPING-REFS>
178 <EXCLUSIVE-AREAS>
179 <EXCLUSIVE-AREA UUID="263a9d1a-8028-5c86-2c8b-f976f106a6be">
180 <SHORT-NAME>EA_OdoValue</SHORT-NAME>
181 </EXCLUSIVE-AREA>
182 </EXCLUSIVE-AREAS>
183 <EVENTS>
184 <TIMING-EVENT UUID="e0808fa2-c0ac-4e89-9494-06d9ab5a03bc">
185 <SHORT-NAME>Event_Runnable1</SHORT-NAME>
186 <START-ON-EVENT-REF DEST="RUNNABLE-
ENTITY"/>/pkg/swc/ASWC/ASWC/FilterInputRunnable</START-ON-EVENT-REF>
187 <PERIOD>0.001</PERIOD>
188 </TIMING-EVENT>
189 <OPERATION-INVOKED-EVENT UUID="13dd6916-3da0-4e2b-b531-c5e6d44138d9">
190 <SHORT-NAME>opInvokedEv</SHORT-NAME>
191 <START-ON-EVENT-REF DEST="RUNNABLE-
ENTITY"/>/pkg/swc/ASWC/ASWC/OverSpeedDiagnostic</START-ON-EVENT-REF>
192 <OPERATION-IREF>
193 <CONTEXT-P-PORT-REF DEST="P-PORT-
PROTOTYPE"/>/pkg/swc/ASWC/DiagnosticPort</CONTEXT-P-PORT-REF>
194 <TARGET-PROVIDED-OPERATION-REF DEST="CLIENT-SERVER-
OPERATION"/>/pkg/swc/DiagnosticInterface/DataRequest</TARGET-PROVIDED-
OPERATION-REF>
195 </OPERATION-IREF>
196 </OPERATION-INVOKED-EVENT>
197 <TIMING-EVENT UUID="267ae05e-8b57-4733-82d9-379e6ce31a72">
198 <SHORT-NAME>Event_Runnable2</SHORT-NAME>
199 <START-ON-EVENT-REF DEST="RUNNABLE-
ENTITY"/>/pkg/swc/ASWC/ASWC/UpdateOdometerRunnable</START-ON-EVENT-
REF>
200 <PERIOD>0.001</PERIOD>

```

```

201 </TIMING-EVENT>
202 <SWC-MODE-SWITCH-EVENT UUID="5a054f0c-7d28-5d05-e951-e68fc16436d4">
203 <SHORT-NAME>WriteNvmAtSleep</SHORT-NAME>
204 <START-ON-EVENT-REF DEST="RUNNABLE-
ENTITY">/pkg/swc/ASWC/ASWC/StoreOdometerBeforeSleep</START-ON-EVENT-
REF>
205 <ACTIVATION>ON-ENTRY</ACTIVATION>
206 <MODE-IREFS>
207 <MODE-IREF>
208 <CONTEXT-PORT-REF DEST="R-PORT-
PROTOTYPE">/pkg/swc/ASWC/PowerModePort</CONTEXT-PORT-REF>
209 <CONTEXT-MODE-DECLARATION-GROUP-PROTOTYPE-REF DEST="MODE-
DECLARATION-GROUP-
PROTOTYPE">/Modes/PowerModeInterface/PowerModeData</CONTEXT-MODE-
DECLARATION-GROUP-PROTOTYPE-REF>
210 <TARGET-MODE-DECLARATION-REF DEST="MODE-
DECLARATION">/Modes/PowerModeGroup/SLEEP</TARGET-MODE-DECLARATION-
REF>
211 </MODE-IREF>
212 </MODE-IREFS>
213 </SWC-MODE-SWITCH-EVENT>
214 </EVENTS>
215 <HANDLE-TERMINATION-AND-RESTART>NO-SUPPORT</HANDLE-TERMINATION-
AND-RESTART>
216 <IMPLICIT-INTER-RUNNABLE-VARIABLES>
217 <VARIABLE-DATA-PROTOTYPE UUID="70c777fb-20e5-44ab-aa4b-0b9e59580165">
218 <SHORT-NAME>currentodo</SHORT-NAME>
219 <CATEGORY>VALUE</CATEGORY>
220 <SW-DATA-DEF-PROPS>
221 <SW-DATA-DEF-PROPS-VARIANTS>
222 <SW-DATA-DEF-PROPS-CONDITIONAL>
223 <SW-CALIBRATION-ACCESS>NOT-ACCESSIBLE</SW-CALIBRATION-ACCESS>
224 <SW-IMPL-POLICY>STANDARD</SW-IMPL-POLICY>
225 </SW-DATA-DEF-PROPS-CONDITIONAL>
226 </SW-DATA-DEF-PROPS-VARIANTS>
227 </SW-DATA-DEF-PROPS>
228 <TYPE-TREF DEST="IMPLEMENTATION-DATA-TYPE">/pkg/ASWC_dt/UInt16</TYPE-
TREF>
229 <INIT-VALUE>
230 <CONSTANT-REFERENCE>
231 <SHORT-LABEL>DefaultInitValue_UInt16</SHORT-LABEL>

```

```

232 <CONSTANT-REF DEST="CONSTANT-
SPECIFICATION"/>/pkg/ASWC_dt/Ground/DefaultInitValue_UInt16</CONSTANT-REF>
233 </CONSTANT-REFERENCE>
234 </INIT-VALUE>
235 </VARIABLE-DATA-PROTOTYPE>
236 <VARIABLE-DATA-PROTOTYPE UUID="ac799b1b-7c1a-4e11-bcd2-3fbda41981c4">
237 <SHORT-NAME>pulsedata</SHORT-NAME>
238 <CATEGORY>VALUE</CATEGORY>
239 <SW-DATA-DEF-PROPS>
240 <SW-DATA-DEF-PROPS-VARIANTS>
241 <SW-DATA-DEF-PROPS-CONDITIONAL>
242 <SW-CALIBRATION-ACCESS>NOT-ACCESSIBLE</SW-CALIBRATION-ACCESS>
243 <SW-IMPL-POLICY>STANDARD</SW-IMPL-POLICY>
244 </SW-DATA-DEF-PROPS-CONDITIONAL>
245 </SW-DATA-DEF-PROPS-VARIANTS>
246 </SW-DATA-DEF-PROPS>
247 <TYPE-TREF DEST="IMPLEMENTATION-DATA-TYPE"/>/pkg/ASWC_dt/UInt8</TYPE-
TREF>
248 <INIT-VALUE>
249 <CONSTANT-REFERENCE>
250 <SHORT-LABEL>DefaultInitValue_UInt8</SHORT-LABEL>
251 <CONSTANT-REF DEST="CONSTANT-
SPECIFICATION"/>/pkg/ASWC_dt/Ground/DefaultInitValue_UInt8</CONSTANT-REF>
252 </CONSTANT-REFERENCE>
253 </INIT-VALUE>
254 </VARIABLE-DATA-PROTOTYPE>
255 <VARIABLE-DATA-PROTOTYPE UUID="ebf95658-6e6f-4271-99bc-a041d1f51dc1">
256 <SHORT-NAME>vehiclespeed</SHORT-NAME>
257 <CATEGORY>VALUE</CATEGORY>
258 <SW-DATA-DEF-PROPS>
259 <SW-DATA-DEF-PROPS-VARIANTS>
260 <SW-DATA-DEF-PROPS-CONDITIONAL>
261 <SW-CALIBRATION-ACCESS>NOT-ACCESSIBLE</SW-CALIBRATION-ACCESS>
262 <SW-IMPL-POLICY>STANDARD</SW-IMPL-POLICY>
263 </SW-DATA-DEF-PROPS-CONDITIONAL>
264 </SW-DATA-DEF-PROPS-VARIANTS>
265 </SW-DATA-DEF-PROPS>
266 <TYPE-TREF DEST="IMPLEMENTATION-DATA-
TYPE"/>/pkg/ASWC_dt/Double</TYPE-TREF>

```

```
267 <INIT-VALUE>
268 <CONSTANT-REFERENCE>
269 <SHORT-LABEL>DefaultInitValue_Double</SHORT-LABEL>
270 <CONSTANT-REF DEST="CONSTANT-
SPECIFICATION"/>/pkg/ASWC_dt/Ground/DefaultInitValue_Double</CONSTANT-REF>
271 </CONSTANT-REFERENCE>
272 </INIT-VALUE>
273 </VARIABLE-DATA-PROTOTYPE>
274 </IMPLICIT-INTER-RUNNABLE-VARIABLES>
275 <PER-INSTANCE-MEMORYS>
276 <PER-INSTANCE-MEMORY UUID="36150b00-0266-5bb3-429b-c0e39b109649">
277 <SHORT-NAME>OdoValue</SHORT-NAME>
278 <INIT-VALUE/>
279 <SW-DATA-DEF-PROPS>
280 <SW-DATA-DEF-PROPS-VARIANTS>
281 <SW-DATA-DEF-PROPS-CONDITIONAL/>
282 </SW-DATA-DEF-PROPS-VARIANTS>
283 </SW-DATA-DEF-PROPS>
284 <TYPE>OdoValue_UInt16</TYPE>
285 <TYPE-DEFINITION>UInt16</TYPE-DEFINITION>
286 </PER-INSTANCE-MEMORY>
287 </PER-INSTANCE-MEMORYS>
288 <RUNNABLES>
289 <RUNNABLE-ENTITY UUID="8828da4b-8a28-435c-a613-bee7b2f380f5">
290 <SHORT-NAME>Runnable_Init</SHORT-NAME>
291 <MINIMUM-START-INTERVAL>0</MINIMUM-START-INTERVAL>
292 <CAN-BE-INVOKED-CONCURRENTLY>>false</CAN-BE-INVOKED-CONCURRENTLY>
293 <SYMBOL>Runnable_Init</SYMBOL>
294 </RUNNABLE-ENTITY>
295 <RUNNABLE-ENTITY UUID="9ed44f19-1ef9-4904-aa12-534a437bd7dc">
296 <SHORT-NAME>FilterInputRunnable</SHORT-NAME>
297 <MINIMUM-START-INTERVAL>0</MINIMUM-START-INTERVAL>
298 <CAN-BE-INVOKED-CONCURRENTLY>>false</CAN-BE-INVOKED-CONCURRENTLY>
299 <DATA-READ-ACCESSS>
300 <VARIABLE-ACCESS UUID="812ee79a-35c3-5201-3d11-3746c0b89660">
301 <SHORT-NAME>IN_RPort_Pulse_Data_RPort_Pulse_Data</SHORT-NAME>
302 <ACCESSED-VARIABLE>
303 <AUTOSAR-VARIABLE-IREF>
```

```

304 <PORT-PROTOTYPE-REF                                DEST="R-PORT-
    PROTOTYPE">/pkg/swc/ASWC/RPort_Pulse_Data</PORT-PROTOTYPE-REF>
305 <TARGET-DATA-PROTOTYPE-REF                          DEST="VARIABLE-DATA-
    PROTOTYPE">/pkg/if/RPort_Pulse_Data/RPort_Pulse_Data</TARGET-DATA-
    PROTOTYPE-REF>
306 </AUTOSAR-VARIABLE-IREF>
307 </ACCESSED-VARIABLE>
308 </VARIABLE-ACCESS>
309 <VARIABLE-ACCESS UUID="cfe897fc-c037-54d8-f3e8-0189ec817908">
310 <SHORT-NAME>IN_RPort_Pulse_Status_RPort_Pulse_Status</SHORT-NAME>
311 <ACCESSED-VARIABLE>
312 <AUTOSAR-VARIABLE-IREF>
313 <PORT-PROTOTYPE-REF                                DEST="R-PORT-
    PROTOTYPE">/pkg/swc/ASWC/RPort_Pulse_Status</PORT-PROTOTYPE-REF>
314 <TARGET-DATA-PROTOTYPE-REF                          DEST="VARIABLE-DATA-
    PROTOTYPE">/pkg/if/RPort_Pulse_Status/RPort_Pulse_Status</TARGET-DATA-
    PROTOTYPE-REF>
315 </AUTOSAR-VARIABLE-IREF>
316 </ACCESSED-VARIABLE>
317 </VARIABLE-ACCESS>
318 <VARIABLE-ACCESS UUID="65882d4d-e9c0-5f43-0740-9242bf0e8d74">
319 <SHORT-NAME>IN_WheelSpeedPort_WheelSpeedArray</SHORT-NAME>
320 <ACCESSED-VARIABLE>
321 <AUTOSAR-VARIABLE-IREF>
322 <PORT-PROTOTYPE-REF                                DEST="R-PORT-
    PROTOTYPE">/pkg/swc/ASWC/WheelSpeedPort</PORT-PROTOTYPE-REF>
323 <TARGET-DATA-PROTOTYPE-REF                          DEST="VARIABLE-DATA-
    PROTOTYPE">/pkg/if/RPortWheelSpeedInterface/WheelSpeedArray</TARGET-DATA-
    PROTOTYPE-REF>
324 </AUTOSAR-VARIABLE-IREF>
325 </ACCESSED-VARIABLE>
326 </VARIABLE-ACCESS>
327 </DATA-READ-ACCESS>
328 <DATA-WRITE-ACCESS>
329 <VARIABLE-ACCESS UUID="c4859508-3130-5dc0-bd4f-6a8bc80af76e">
330 <SHORT-NAME>OUT_VehicleSpeedPort_VehicleSpeedData</SHORT-NAME>
331 <ACCESSED-VARIABLE>
332 <AUTOSAR-VARIABLE-IREF>
333 <PORT-PROTOTYPE-REF                                DEST="P-PORT-
    PROTOTYPE">/pkg/swc/ASWC/VehicleSpeedPort</PORT-PROTOTYPE-REF>
334 <TARGET-DATA-PROTOTYPE-REF                          DEST="VARIABLE-DATA-
    PROTOTYPE">/pkg/if/PPortVehicleSpeed/VehicleSpeedData</TARGET-DATA-

```

```

    PROTOTYPE-REF>
335 </AUTOSAR-VARIABLE-IREF>
336 </ACCESSED-VARIABLE>
337 </VARIABLE-ACCESS>
338 </DATA-WRITE-ACCESS>
339 <MODE-ACCESS-POINTS>
340 <MODE-ACCESS-POINT>
341 <MODE-GROUP-IREF>
342 <R-MODE-GROUP-IN-ATOMIC-SWC-INSTANCE-REF>
343 <CONTEXT-R-PORT-REF                                DEST="R-PORT-
    PROTOTYPE">/pkg/swc/ASWC/PowerModePort</CONTEXT-R-PORT-REF>
344 <TARGET-MODE-GROUP-REF                            DEST="MODE-DECLARATION-GROUP-
    PROTOTYPE">/Modes/PowerModeInterface/PowerModeData</TARGET-MODE-
    GROUP-REF>
345 </R-MODE-GROUP-IN-ATOMIC-SWC-INSTANCE-REF>
346 </MODE-GROUP-IREF>
347 </MODE-ACCESS-POINT>
348 </MODE-ACCESS-POINTS>
349 <SYMBOL>FilterInputRunnable</SYMBOL>
350 <WRITTEN-LOCAL-VARIABLES>
351 <VARIABLE-ACCESS UUID="23268391-1c7f-54ea-d8b9-597be415c165">
352 <SHORT-NAME>WV_pulsedata</SHORT-NAME>
353 <ACCESSED-VARIABLE>
354 <LOCAL-VARIABLE-REF                                DEST="VARIABLE-DATA-
    PROTOTYPE">/pkg/swc/ASWC/ASWC/pulsedata</LOCAL-VARIABLE-REF>
355 </ACCESSED-VARIABLE>
356 </VARIABLE-ACCESS>
357 <VARIABLE-ACCESS UUID="c2af303b-c5c5-5fd2-9c1b-1b5606761b8a">
358 <SHORT-NAME>WV_vehiclespeed</SHORT-NAME>
359 <ACCESSED-VARIABLE>
360 <LOCAL-VARIABLE-REF                                DEST="VARIABLE-DATA-
    PROTOTYPE">/pkg/swc/ASWC/ASWC/vehiclespeed</LOCAL-VARIABLE-REF>
361 </ACCESSED-VARIABLE>
362 </VARIABLE-ACCESS>
363 </WRITTEN-LOCAL-VARIABLES>
364 </RUNNABLE-ENTITY>
365 <RUNNABLE-ENTITY UUID="d87f1965-7d40-47c8-8ecb-e9d2322bb007">
366 <SHORT-NAME>OverSpeedDiagnostic</SHORT-NAME>
367 <MINIMUM-START-INTERVAL>0</MINIMUM-START-INTERVAL>

```



```

368 <CAN-BE-INVOKED-CONCURRENTLY>false</CAN-BE-INVOKED-CONCURRENTLY>
369 <DATA-READ-ACCESSS>
370 <VARIABLE-ACCESS UUID="40f75788-6672-5e49-8007-c0a382aaa495">
371 <SHORT-NAME>IN_RPort_Pulse_Status_RPort_Pulse_Status1</SHORT-NAME>
372 <ACCESSED-VARIABLE>
373 <AUTOSAR-VARIABLE-IREF>
374 <PORT-PROTOTYPE-REF DEST="R-PORT-
    PROTOTYPE"/>/pkg/swc/ASWC/RPort_Pulse_Status</PORT-PROTOTYPE-REF>
375 <TARGET-DATA-PROTOTYPE-REF DEST="VARIABLE-DATA-
    PROTOTYPE"/>/pkg/if/RPort_Pulse_Status/RPort_Pulse_Status1</TARGET-DATA-
    PROTOTYPE-REF>
376 </AUTOSAR-VARIABLE-IREF>
377 </ACCESSED-VARIABLE>
378 </VARIABLE-ACCESS>
379 </DATA-READ-ACCESSS>
380 <DATA-WRITE-ACCESSS>
381 <VARIABLE-ACCESS UUID="83473742-fb78-58e0-e0e9-4de290cac4ac">
382 <SHORT-NAME>OUT_PPort_DE5_PPort_Status</SHORT-NAME>
383 <ACCESSED-VARIABLE>
384 <AUTOSAR-VARIABLE-IREF>
385 <PORT-PROTOTYPE-REF DEST="P-PORT-
    PROTOTYPE"/>/pkg/swc/ASWC/PPort_DE5</PORT-PROTOTYPE-REF>
386 <TARGET-DATA-PROTOTYPE-REF DEST="VARIABLE-DATA-
    PROTOTYPE"/>/pkg/if/PPort_Status/PPort_Status</TARGET-DATA-PROTOTYPE-
    REF>
387 </AUTOSAR-VARIABLE-IREF>
388 </ACCESSED-VARIABLE>
389 </VARIABLE-ACCESS>
390 </DATA-WRITE-ACCESSS>
391 <READ-LOCAL-VARIABLES>
392 <VARIABLE-ACCESS UUID="fca73c64-dadd-593d-598f-39337b645b40">
393 <SHORT-NAME>RV_vehiclespeed</SHORT-NAME>
394 <ACCESSED-VARIABLE>
395 <LOCAL-VARIABLE-REF DEST="VARIABLE-DATA-
    PROTOTYPE"/>/pkg/swc/ASWC/ASWC/vehiclespeed</LOCAL-VARIABLE-REF>
396 </ACCESSED-VARIABLE>
397 </VARIABLE-ACCESS>
398 </READ-LOCAL-VARIABLES>
399 <SYMBOL>OverSpeedDiagnostic</SYMBOL>
400 </RUNNABLE-ENTITY>

```

```

401 <RUNNABLE-ENTITY UUID="a866c8bb-761d-55d1-7d5f-2f95ef5af16f">
402 <SHORT-NAME>StoreOdometerBeforeSleep</SHORT-NAME>
403 <MINIMUM-START-INTERVAL>0</MINIMUM-START-INTERVAL>
404 <RUNS-INSIDE-EXCLUSIVE-AREA-REFS>
405 <RUNS-INSIDE-EXCLUSIVE-AREA-REF DEST="EXCLUSIVE-
AREA">/pkg/swc/ASWC/ASWC/EA_OdoValue</RUNS-INSIDE-EXCLUSIVE-AREA-
REF>
406 </RUNS-INSIDE-EXCLUSIVE-AREA-REFS>
407 <CAN-BE-INVOKED-CONCURRENTLY>>false</CAN-BE-INVOKED-CONCURRENTLY>
408 <READ-LOCAL-VARIABLES>
409 <VARIABLE-ACCESS UUID="db89561b-f0d6-53f1-4e3a-24045d2c76e6">
410 <SHORT-NAME>RV_currentodo</SHORT-NAME>
411 <ACCESSED-VARIABLE>
412 <LOCAL-VARIABLE-REF DEST="VARIABLE-DATA-
PROTOTYPE">/pkg/swc/ASWC/ASWC/currentodo</LOCAL-VARIABLE-REF>
413 </ACCESSED-VARIABLE>
414 </VARIABLE-ACCESS>
415 </READ-LOCAL-VARIABLES>
416 <SERVER-CALL-POINTS>
417 <SYNCHRONOUS-SERVER-CALL-POINT UUID="0e7552ac-9720-57ec-9fd9-
4c66d8c832f7">
418 <SHORT-NAME>SC_NvmService_SetRamBlockStatus</SHORT-NAME>
419 <OPERATION-IREF>
420 <CONTEXT-R-PORT-REF DEST="R-PORT-
PROTOTYPE">/pkg/swc/ASWC/NvmService</CONTEXT-R-PORT-REF>
421 <TARGET-REQUIRED-OPERATION-REF DEST="CLIENT-SERVER-
OPERATION">/AUTOSAR/Services/NvM/SetRamBlockStatus</TARGET-REQUIRED-
OPERATION-REF>
422 </OPERATION-IREF>
423 <TIMEOUT>1.0e-06</TIMEOUT>
424 </SYNCHRONOUS-SERVER-CALL-POINT>
425 </SERVER-CALL-POINTS>
426 <SYMBOL>StoreOdometerBeforeSleep</SYMBOL>
427 </RUNNABLE-ENTITY>
428 <RUNNABLE-ENTITY UUID="2d5dca89-0e24-46d7-bee3-e9318435b713">
429 <SHORT-NAME>UpdateOdometerRunnable</SHORT-NAME>
430 <MINIMUM-START-INTERVAL>0</MINIMUM-START-INTERVAL>
431 <RUNS-INSIDE-EXCLUSIVE-AREA-REFS>
432 <RUNS-INSIDE-EXCLUSIVE-AREA-REF DEST="EXCLUSIVE-
AREA">/pkg/swc/ASWC/ASWC/EA_OdoValue</RUNS-INSIDE-EXCLUSIVE-AREA-
REF>

```

```

433 </RUNS-INSIDE-EXCLUSIVE-AREA-REFS>
434 <CAN-BE-INVOKED-CONCURRENTLY>false</CAN-BE-INVOKED-CONCURRENTLY>
435 <DATA-WRITE-ACCESSS>
436 <VARIABLE-ACCESS UUID="0af7bcdd-bbf2-54e5-40ea-91d1fab52abf">
437 <SHORT-NAME>OUT_PPport_OdoValue_OdoValue</SHORT-NAME>
438 <ACCESSED-VARIABLE>
439 <AUTOSAR-VARIABLE-IREF>
440 <PORT-PROTOTYPE-REF DEST="P-PORT-
    PROTOTYPE"/>pkg/swc/ASWC/PPort_OdoValue</PORT-PROTOTYPE-REF>
441 <TARGET-DATA-PROTOTYPE-REF DEST="VARIABLE-DATA-
    PROTOTYPE"/>pkg/if/OdoValue/OdoValue</TARGET-DATA-PROTOTYPE-REF>
442 </AUTOSAR-VARIABLE-IREF>
443 </ACCESSED-VARIABLE>
444 </VARIABLE-ACCESS>
445 </DATA-WRITE-ACCESSS>
446 <READ-LOCAL-VARIABLES>
447 <VARIABLE-ACCESS UUID="6c11fa21-75e4-51f0-6a07-e73dca1689f9">
448 <SHORT-NAME>RV_pulsedata</SHORT-NAME>
449 <ACCESSED-VARIABLE>
450 <LOCAL-VARIABLE-REF DEST="VARIABLE-DATA-
    PROTOTYPE"/>pkg/swc/ASWC/ASWC/pulsedata</LOCAL-VARIABLE-REF>
451 </ACCESSED-VARIABLE>
452 </VARIABLE-ACCESS>
453 </READ-LOCAL-VARIABLES>
454 <SYMBOL>UpdateOdometerRunnable</SYMBOL>
455 <WRITTEN-LOCAL-VARIABLES>
456 <VARIABLE-ACCESS UUID="4d604d6e-6bbc-53ad-b3cd-4d63b04c770c">
457 <SHORT-NAME>WV_currentodo</SHORT-NAME>
458 <ACCESSED-VARIABLE>
459 <LOCAL-VARIABLE-REF DEST="VARIABLE-DATA-
    PROTOTYPE"/>pkg/swc/ASWC/ASWC/currentodo</LOCAL-VARIABLE-REF>
460 </ACCESSED-VARIABLE>
461 </VARIABLE-ACCESS>
462 </WRITTEN-LOCAL-VARIABLES>
463 </RUNNABLE-ENTITY>
464 </RUNNABLES>
465 <SUPPORTS-MULTIPLE-INSTANTIATION>false</SUPPORTS-MULTIPLE-
    INSTANTIATION>
466 </SWC-INTERNAL-BEHAVIOR>

```

```
467 </INTERNAL-BEHAVIORS>
468 </APPLICATION-SW-COMPONENT-TYPE>
469 <CLIENT-SERVER-INTERFACE UUID="5c7d5f50-cb0d-4ad5-8994-bee75714e815">
470 <SHORT-NAME>DiagnosticInterface</SHORT-NAME>
471 <IS-SERVICE>>false</IS-SERVICE>
472 <OPERATIONS>
473 <CLIENT-SERVER-OPERATION UUID="e034c12f-1dfc-400d-8c70-302dd214f63d">
474 <SHORT-NAME>DataRequest</SHORT-NAME>
475 <ARGUMENTS>
476 <ARGUMENT-DATA-PROTOTYPE UUID="28d92d9c-296d-53ec-7c5e-32e433d9fc23">
477 <SHORT-NAME>PortArg</SHORT-NAME>
478 <CATEGORY>VALUE</CATEGORY>
479 <SW-DATA-DEF-PROPS>
480 <SW-DATA-DEF-PROPS-VARIANTS>
481 <SW-DATA-DEF-PROPS-CONDITIONAL>
482 <SW-CALIBRATION-ACCESS>READ-ONLY</SW-CALIBRATION-ACCESS>
483 <SW-IMPL-POLICY>STANDARD</SW-IMPL-POLICY>
484 </SW-DATA-DEF-PROPS-CONDITIONAL>
485 </SW-DATA-DEF-PROPS-VARIANTS>
486 </SW-DATA-DEF-PROPS>
487 <TYPE-TREF DEST="IMPLEMENTATION-DATA-TYPE"/>/pkg/ASWC_dt/UInt8</TYPE-
TREF>
488 <DIRECTION>IN</DIRECTION>
489 </ARGUMENT-DATA-PROTOTYPE>
490 </ARGUMENTS>
491 </CLIENT-SERVER-OPERATION>
492 </OPERATIONS>
493 </CLIENT-SERVER-INTERFACE>
494 </ELEMENTS>
495 </AR-PACKAGE>
496 <AR-PACKAGE UUID="b2ec08f7-3d70-5839-c07b-7a7d74ccf83d">
497 <SHORT-NAME>if</SHORT-NAME>
498 <ELEMENTS>
499 <SENDER-RECEIVER-INTERFACE UUID="1425ca29-a309-4927-84b7-2feb8d1abf4d">
500 <SHORT-NAME>OdoValue</SHORT-NAME>
501 <IS-SERVICE>>false</IS-SERVICE>
502 <DATA-ELEMENTS>
503 <VARIABLE-DATA-PROTOTYPE UUID="3e14fc0c-81ef-4f35-aeac-f38817664baa">
```

```
504 <SHORT-NAME>OdoValue</SHORT-NAME>
505 <CATEGORY>VALUE</CATEGORY>
506 <SW-DATA-DEF-PROPS>
507 <SW-DATA-DEF-PROPS-VARIANTS>
508 <SW-DATA-DEF-PROPS-CONDITIONAL>
509 <SW-CALIBRATION-ACCESS>NOT-ACCESSIBLE</SW-CALIBRATION-ACCESS>
510 <SW-IMPL-POLICY>STANDARD</SW-IMPL-POLICY>
511 </SW-DATA-DEF-PROPS-CONDITIONAL>
512 </SW-DATA-DEF-PROPS-VARIANTS>
513 </SW-DATA-DEF-PROPS>
514 <TYPE-TREF DEST="IMPLEMENTATION-DATA-TYPE">/pkg/ASWC_dt/UInt16</TYPE-
TREF>
515 </VARIABLE-DATA-PROTOTYPE>
516 </DATA-ELEMENTS>
517 </SENDER-RECEIVER-INTERFACE>
518 <SENDER-RECEIVER-INTERFACE UUID="f1351b4b-2a2f-4f1f-93db-d598addce323">
519 <SHORT-NAME>PPort_Status</SHORT-NAME>
520 <IS-SERVICE>>false</IS-SERVICE>
521 <DATA-ELEMENTS>
522 <VARIABLE-DATA-PROTOTYPE UUID="8920275f-2464-4255-9595-3935fab32445">
523 <SHORT-NAME>PPort_Status</SHORT-NAME>
524 <CATEGORY>VALUE</CATEGORY>
525 <SW-DATA-DEF-PROPS>
526 <SW-DATA-DEF-PROPS-VARIANTS>
527 <SW-DATA-DEF-PROPS-CONDITIONAL>
528 <SW-CALIBRATION-ACCESS>NOT-ACCESSIBLE</SW-CALIBRATION-ACCESS>
529 <SW-IMPL-POLICY>STANDARD</SW-IMPL-POLICY>
530 </SW-DATA-DEF-PROPS-CONDITIONAL>
531 </SW-DATA-DEF-PROPS-VARIANTS>
532 </SW-DATA-DEF-PROPS>
533 <TYPE-TREF DEST="IMPLEMENTATION-DATA-
TYPE">/pkg/ASWC_dt/Boolean</TYPE-TREF>
534 </VARIABLE-DATA-PROTOTYPE>
535 </DATA-ELEMENTS>
536 </SENDER-RECEIVER-INTERFACE>
537 <SENDER-RECEIVER-INTERFACE UUID="4e40c115-f77c-46f7-b787-111c828b7c09">
538 <SHORT-NAME>RPort_Pulse_Data</SHORT-NAME>
539 <IS-SERVICE>>false</IS-SERVICE>
```

```
540 <DATA-ELEMENTS>
541 <VARIABLE-DATA-PROTOTYPE UUID="31698255-c94a-4140-b0ca-a1debd23650c">
542 <SHORT-NAME>RPort_Pulse_Data</SHORT-NAME>
543 <CATEGORY>VALUE</CATEGORY>
544 <SW-DATA-DEF-PROPS>
545 <SW-DATA-DEF-PROPS-VARIANTS>
546 <SW-DATA-DEF-PROPS-CONDITIONAL>
547 <SW-CALIBRATION-ACCESS>NOT-ACCESSIBLE</SW-CALIBRATION-ACCESS>
548 <SW-IMPL-POLICY>STANDARD</SW-IMPL-POLICY>
549 </SW-DATA-DEF-PROPS-CONDITIONAL>
550 </SW-DATA-DEF-PROPS-VARIANTS>
551 </SW-DATA-DEF-PROPS>
552 <TYPE-TREF DEST="IMPLEMENTATION-DATA-TYPE">/pkg/ASWC_dt/UInt8</TYPE-
TREF>
553 </VARIABLE-DATA-PROTOTYPE>
554 </DATA-ELEMENTS>
555 </SENDER-RECEIVER-INTERFACE>
556 <SENDER-RECEIVER-INTERFACE UUID="b809d51f-9448-4c68-9649-8a22901f3f75">
557 <SHORT-NAME>RPort_Pulse_Status</SHORT-NAME>
558 <IS-SERVICE>>false</IS-SERVICE>
559 <DATA-ELEMENTS>
560 <VARIABLE-DATA-PROTOTYPE UUID="125151a4-1674-4c4e-aeab-1588a4578409">
561 <SHORT-NAME>RPort_Pulse_Status</SHORT-NAME>
562 <CATEGORY>VALUE</CATEGORY>
563 <SW-DATA-DEF-PROPS>
564 <SW-DATA-DEF-PROPS-VARIANTS>
565 <SW-DATA-DEF-PROPS-CONDITIONAL>
566 <SW-CALIBRATION-ACCESS>NOT-ACCESSIBLE</SW-CALIBRATION-ACCESS>
567 <SW-IMPL-POLICY>STANDARD</SW-IMPL-POLICY>
568 </SW-DATA-DEF-PROPS-CONDITIONAL>
569 </SW-DATA-DEF-PROPS-VARIANTS>
570 </SW-DATA-DEF-PROPS>
571 <TYPE-TREF DEST="IMPLEMENTATION-DATA-
TYPE">/pkg/ASWC_dt/Boolean</TYPE-TREF>
572 </VARIABLE-DATA-PROTOTYPE>
573 <VARIABLE-DATA-PROTOTYPE UUID="f36c0877-b822-5b52-b3cd-8ae8284ff15d">
574 <SHORT-NAME>RPort_Pulse_Status1</SHORT-NAME>
575 <CATEGORY>VALUE</CATEGORY>
```

```
576 <SW-DATA-DEF-PROPS>
577 <SW-DATA-DEF-PROPS-VARIANTS>
578 <SW-DATA-DEF-PROPS-CONDITIONAL>
579 <SW-CALIBRATION-ACCESS>NOT-ACCESSIBLE</SW-CALIBRATION-ACCESS>
580 <SW-IMPL-POLICY>STANDARD</SW-IMPL-POLICY>
581 </SW-DATA-DEF-PROPS-CONDITIONAL>
582 </SW-DATA-DEF-PROPS-VARIANTS>
583 </SW-DATA-DEF-PROPS>
584 <TYPE-TREF                                DEST="IMPLEMENTATION-DATA-
    TYPE"/>/pkg/ASWC_dt/Boolean</TYPE-TREF>
585 </VARIABLE-DATA-PROTOTYPE>
586 </DATA-ELEMENTS>
587 </SENDER-RECEIVER-INTERFACE>
588 <CLIENT-SERVER-INTERFACE UUID="6f0bbc5d-03d9-4d22-921b-bbf5d5a98073">
589 <SHORT-NAME>NvmInterface</SHORT-NAME>
590 <IS-SERVICE>>false</IS-SERVICE>
591 <OPERATIONS>
592 <CLIENT-SERVER-OPERATION UUID="d9d009d4-4064-4571-8082-d4682570c44a">
593 <SHORT-NAME>NvmWrite</SHORT-NAME>
594 </CLIENT-SERVER-OPERATION>
595 </OPERATIONS>
596 </CLIENT-SERVER-INTERFACE>
597 <SENDER-RECEIVER-INTERFACE UUID="eff63e9a-db19-4612-9f76-bff7f04d17a5">
598 <SHORT-NAME>RPortWheelSpeedInterface</SHORT-NAME>
599 <IS-SERVICE>>false</IS-SERVICE>
600 <DATA-ELEMENTS>
601 <VARIABLE-DATA-PROTOTYPE UUID="650bca71-4993-4c81-b83c-63cc79a9a421">
602 <SHORT-NAME>WheelSpeedArray</SHORT-NAME>
603 <CATEGORY>ARRAY</CATEGORY>
604 <SW-DATA-DEF-PROPS>
605 <SW-DATA-DEF-PROPS-VARIANTS>
606 <SW-DATA-DEF-PROPS-CONDITIONAL>
607 <SW-CALIBRATION-ACCESS>NOT-ACCESSIBLE</SW-CALIBRATION-ACCESS>
608 <SW-IMPL-POLICY>STANDARD</SW-IMPL-POLICY>
609 </SW-DATA-DEF-PROPS-CONDITIONAL>
610 </SW-DATA-DEF-PROPS-VARIANTS>
611 </SW-DATA-DEF-PROPS>
612 <TYPE-TREF                                DEST="IMPLEMENTATION-DATA-
```

```

        TYPE"/>pkg/ASWC_dt/rt_Array_Double_4</TYPE-TREF>
613 </VARIABLE-DATA-PROTOTYPE>
614 </DATA-ELEMENTS>
615 </SENDER-RECEIVER-INTERFACE>
616 <SENDER-RECEIVER-INTERFACE UUID="186a1efc-972e-48da-8c90-0c3953c48391">
617 <SHORT-NAME>PPortVehicleSpeed</SHORT-NAME>
618 <IS-SERVICE>>false</IS-SERVICE>
619 <DATA-ELEMENTS>
620 <VARIABLE-DATA-PROTOTYPE UUID="e6beae18-6f6a-4189-91bc-286af40a2fb7">
621 <SHORT-NAME>VehicleSpeedData</SHORT-NAME>
622 <CATEGORY>VALUE</CATEGORY>
623 <SW-DATA-DEF-PROPS>
624 <SW-DATA-DEF-PROPS-VARIANTS>
625 <SW-DATA-DEF-PROPS-CONDITIONAL>
626 <SW-CALIBRATION-ACCESS>NOT-ACCESSIBLE</SW-CALIBRATION-ACCESS>
627 <SW-IMPL-POLICY>STANDARD</SW-IMPL-POLICY>
628 </SW-DATA-DEF-PROPS-CONDITIONAL>
629 </SW-DATA-DEF-PROPS-VARIANTS>
630 </SW-DATA-DEF-PROPS>
631 <TYPE-TREF                                DEST="IMPLEMENTATION-DATA-
        TYPE"/>pkg/ASWC_dt/Double</TYPE-TREF>
632 </VARIABLE-DATA-PROTOTYPE>
633 </DATA-ELEMENTS>
634 </SENDER-RECEIVER-INTERFACE>
635 </ELEMENTS>
636 </AR-PACKAGE>
637 <AR-PACKAGE UUID="46a597d4-84c8-5fd9-15fb-1e71e95ad9e5">
638 <SHORT-NAME>ASWC_dt</SHORT-NAME>
639 <ELEMENTS>
640 <IMPLEMENTATION-DATA-TYPE UUID="83120aa2-edfd-5deb-189a-cbf2ef5077fa">
641 <SHORT-NAME>UInt8</SHORT-NAME>
642 <CATEGORY>VALUE</CATEGORY>
643 <SW-DATA-DEF-PROPS>
644 <SW-DATA-DEF-PROPS-VARIANTS>
645 <SW-DATA-DEF-PROPS-CONDITIONAL>
646 <BASE-TYPE-REF                                DEST="SW-BASE-
        TYPE"/>pkg/ASWC_dt/SwBaseTypes/uint8</BASE-TYPE-REF>
647 <DATA-CONSTR-REF                                DEST="DATA-
        CONSTR"/>pkg/ASWC_dt/DataConstrs/DC_UInt8</DATA-CONSTR-REF>

```



```

648 </SW-DATA-DEF-PROPS-CONDITIONAL>
649 </SW-DATA-DEF-PROPS-VARIANTS>
650 </SW-DATA-DEF-PROPS>
651 </IMPLEMENTATION-DATA-TYPE>
652 <IMPLEMENTATION-DATA-TYPE UUID="2e29a68c-8b53-5cf9-9805-f1ad6db56ca4">
653 <SHORT-NAME>Boolean</SHORT-NAME>
654 <CATEGORY>VALUE</CATEGORY>
655 <SW-DATA-DEF-PROPS>
656 <SW-DATA-DEF-PROPS-VARIANTS>
657 <SW-DATA-DEF-PROPS-CONDITIONAL>
658 <BASE-TYPE-REF DEST="SW-BASE-
TYPE"/>/pkg/ASWC_dt/SwBaseTypes/boolean</BASE-TYPE-REF>
659 <COMPU-METHOD-REF DEST="COMPU-
METHOD"/>/pkg/ASWC_dt/COMPU_Boolean</COMPU-METHOD-REF>
660 <DATA-CONSTR-REF DEST="DATA-
CONSTR"/>/pkg/ASWC_dt/DataConstrs/DC_Boolean</DATA-CONSTR-REF>
661 </SW-DATA-DEF-PROPS-CONDITIONAL>
662 </SW-DATA-DEF-PROPS-VARIANTS>
663 </SW-DATA-DEF-PROPS>
664 </IMPLEMENTATION-DATA-TYPE>
665 <COMPU-METHOD UUID="79b8d7ec-c2b4-595a-9b96-845ed9c40abb">
666 <SHORT-NAME>COMPU_Boolean</SHORT-NAME>
667 <CATEGORY>TEXTTABLE</CATEGORY>
668 <UNIT-REF DEST="UNIT"/>/pkg/ASWC_dt/NoUnit</UNIT-REF>
669 <COMPU-INTERNAL-TO-PHYS>
670 <COMPU-SCALES>
671 <COMPU-SCALE>
672 <LOWER-LIMIT INTERVAL-TYPE="CLOSED">0</LOWER-LIMIT>
673 <UPPER-LIMIT INTERVAL-TYPE="CLOSED">0</UPPER-LIMIT>
674 <COMPU-CONST>
675 <VT>FALSE</VT>
676 </COMPU-CONST>
677 </COMPU-SCALE>
678 <COMPU-SCALE>
679 <LOWER-LIMIT INTERVAL-TYPE="CLOSED">1</LOWER-LIMIT>
680 <UPPER-LIMIT INTERVAL-TYPE="CLOSED">1</UPPER-LIMIT>
681 <COMPU-CONST>
682 <VT>TRUE</VT>
683 </COMPU-CONST>

```

```

684 </COMPU-SCALE>
685 </COMPU-SCALES>
686 </COMPU-INTERNAL-TO-PHYS>
687 </COMPU-METHOD>
688 <UNIT UUID="fbd8a914-9fe2-552c-5d51-0d82ac594046">
689 <SHORT-NAME>NoUnit</SHORT-NAME>
690 <FACTOR-SI-TO-UNIT>1</FACTOR-SI-TO-UNIT>
691 <OFFSET-SI-TO-UNIT>0</OFFSET-SI-TO-UNIT>
692 </UNIT>
693 <IMPLEMENTATION-DATA-TYPE UUID="d008034c-8fdc-5478-cc4d-4f466649b47d">
694 <SHORT-NAME>Double</SHORT-NAME>
695 <CATEGORY>VALUE</CATEGORY>
696 <SW-DATA-DEF-PROPS>
697 <SW-DATA-DEF-PROPS-VARIANTS>
698 <SW-DATA-DEF-PROPS-CONDITIONAL>
699 <BASE-TYPE-REF DEST="SW-BASE-
TYPE">/pkg/ASWC_dt/SwBaseTypes/float64</BASE-TYPE-REF>
700 <DATA-CONSTR-REF DEST="DATA-
CONSTR">/pkg/ASWC_dt/DataConstrs/DC_Double</DATA-CONSTR-REF>
701 </SW-DATA-DEF-PROPS-CONDITIONAL>
702 </SW-DATA-DEF-PROPS-VARIANTS>
703 </SW-DATA-DEF-PROPS>
704 </IMPLEMENTATION-DATA-TYPE>
705 <IMPLEMENTATION-DATA-TYPE UUID="760adfe5-b9ef-5178-4925-1b285e6cf1d6">
706 <SHORT-NAME>rt_Array_Double_4</SHORT-NAME>
707 <CATEGORY>ARRAY</CATEGORY>
708 <SUB-ELEMENTS>
709 <IMPLEMENTATION-DATA-TYPE-ELEMENT UUID="ef8e718f-5e27-5c94-3548-
c0df3fbb299f">
710 <SHORT-NAME>rt_Array_Double_4</SHORT-NAME>
711 <CATEGORY>TYPE_REFERENCE</CATEGORY>
712 <ARRAY-SIZE>4</ARRAY-SIZE>
713 <ARRAY-SIZE-SEMANTICS>FIXED-SIZE</ARRAY-SIZE-SEMANTICS>
714 <SW-DATA-DEF-PROPS>
715 <SW-DATA-DEF-PROPS-VARIANTS>
716 <SW-DATA-DEF-PROPS-CONDITIONAL>
717 <IMPLEMENTATION-DATA-TYPE-REF DEST="IMPLEMENTATION-DATA-
TYPE">/pkg/ASWC_dt/Double</IMPLEMENTATION-DATA-TYPE-REF>
718 </SW-DATA-DEF-PROPS-CONDITIONAL>

```

```
719 </SW-DATA-DEF-PROPS-VARIANTS>
720 </SW-DATA-DEF-PROPS>
721 </IMPLEMENTATION-DATA-TYPE-ELEMENT>
722 </SUB-ELEMENTS>
723 </IMPLEMENTATION-DATA-TYPE>
724 <IMPLEMENTATION-DATA-TYPE UUID="c6587562-4ce8-5e2b-9d48-c47c59f31810">
725 <SHORT-NAME>UInt16</SHORT-NAME>
726 <CATEGORY>VALUE</CATEGORY>
727 <SW-DATA-DEF-PROPS>
728 <SW-DATA-DEF-PROPS-VARIANTS>
729 <SW-DATA-DEF-PROPS-CONDITIONAL>
730 <BASE-TYPE-REF DEST="SW-BASE-
TYPE">/pkg/ASWC_dt/SwBaseTypes/uint16</BASE-TYPE-REF>
731 <DATA-CONSTR-REF DEST="DATA-
CONSTR">/pkg/ASWC_dt/DataConstrs/DC_UInt16</DATA-CONSTR-REF>
732 </SW-DATA-DEF-PROPS-CONDITIONAL>
733 </SW-DATA-DEF-PROPS-VARIANTS>
734 </SW-DATA-DEF-PROPS>
735 </IMPLEMENTATION-DATA-TYPE>
736 <IMPLEMENTATION-DATA-TYPE UUID="a32a1bcb-59c4-5bde-3f25-34f485d6021a">
737 <SHORT-NAME>SInt32</SHORT-NAME>
738 <CATEGORY>VALUE</CATEGORY>
739 <SW-DATA-DEF-PROPS>
740 <SW-DATA-DEF-PROPS-VARIANTS>
741 <SW-DATA-DEF-PROPS-CONDITIONAL>
742 <BASE-TYPE-REF DEST="SW-BASE-
TYPE">/pkg/ASWC_dt/SwBaseTypes/sint32</BASE-TYPE-REF>
743 <DATA-CONSTR-REF DEST="DATA-
CONSTR">/pkg/ASWC_dt/DataConstrs/DC_SInt32</DATA-CONSTR-REF>
744 </SW-DATA-DEF-PROPS-CONDITIONAL>
745 </SW-DATA-DEF-PROPS-VARIANTS>
746 </SW-DATA-DEF-PROPS>
747 </IMPLEMENTATION-DATA-TYPE>
748 </ELEMENTS>
749 <AR-PACKAGES>
750 <AR-PACKAGE UUID="1712648a-ba3c-54de-a7a6-c9f8c6c9005b">
751 <SHORT-NAME>Ground</SHORT-NAME>
752 <ELEMENTS>
753 <CONSTANT-SPECIFICATION UUID="24d9ced3-a7f3-5dff-4647-be4297768beb">
```

```

754 <SHORT-NAME>DefaultInitValue_UInt8</SHORT-NAME>
755 <VALUE-SPEC>
756 <NUMERICAL-VALUE-SPECIFICATION>
757 <SHORT-LABEL>DefaultInitValue_UInt8</SHORT-LABEL>
758 <VALUE>0</VALUE>
759 </NUMERICAL-VALUE-SPECIFICATION>
760 </VALUE-SPEC>
761 </CONSTANT-SPECIFICATION>
762 <CONSTANT-SPECIFICATION UUID="f5a66417-dfb3-5ebb-cb00-75a497bbfa76">
763 <SHORT-NAME>DefaultInitValue_Boolean</SHORT-NAME>
764 <VALUE-SPEC>
765 <NUMERICAL-VALUE-SPECIFICATION>
766 <SHORT-LABEL>DefaultInitValue_Boolean</SHORT-LABEL>
767 <VALUE>0</VALUE>
768 </NUMERICAL-VALUE-SPECIFICATION>
769 </VALUE-SPEC>
770 </CONSTANT-SPECIFICATION>
771 <CONSTANT-SPECIFICATION UUID="d9d1316e-cf06-5926-21f6-860602e95eb2">
772 <SHORT-NAME>DefaultInitValue_rt_Array_Double_4</SHORT-NAME>
773 <VALUE-SPEC>
774 <ARRAY-VALUE-SPECIFICATION>
775 <SHORT-LABEL>DefaultInitValue_rt_Array_Double_4</SHORT-LABEL>
776 <ELEMENTS>
777 <CONSTANT-REFERENCE>
778 <SHORT-LABEL>DefaultInitValue_rt_Array_Double_4_rt_Array_Double_4_1</SHORT-
LABEL>
779 <CONSTANT-REF                                DEST="CONSTANT-
SPECIFICATION">/pkg/ASWC_dt/Ground/DefaultInitValue_Double</CONSTANT-REF>
780 </CONSTANT-REFERENCE>
781 <CONSTANT-REFERENCE>
782 <SHORT-LABEL>DefaultInitValue_rt_Array_Double_4_rt_Array_Double_4_2</SHORT-
LABEL>
783 <CONSTANT-REF                                DEST="CONSTANT-
SPECIFICATION">/pkg/ASWC_dt/Ground/DefaultInitValue_Double</CONSTANT-REF>
784 </CONSTANT-REFERENCE>
785 <CONSTANT-REFERENCE>
786 <SHORT-LABEL>DefaultInitValue_rt_Array_Double_4_rt_Array_Double_4_3</SHORT-
LABEL>
787 <CONSTANT-REF                                DEST="CONSTANT-
SPECIFICATION">/pkg/ASWC_dt/Ground/DefaultInitValue_Double</CONSTANT-REF>

```

```

788 </CONSTANT-REFERENCE>
789 <CONSTANT-REFERENCE>
790 <SHORT-LABEL>DefaultInitValue_rt_Array_Double_4_rt_Array_Double_4_4</SHORT-
    LABEL>
791 <CONSTANT-REF                                DEST="CONSTANT-
    SPECIFICATION"/>/pkg/ASWC_dt/Ground/DefaultInitValue_Double</CONSTANT-REF>
792 </CONSTANT-REFERENCE>
793 </ELEMENTS>
794 </ARRAY-VALUE-SPECIFICATION>
795 </VALUE-SPEC>
796 </CONSTANT-SPECIFICATION>
797 <CONSTANT-SPECIFICATION UUID="4c01e238-5c6f-51fc-f943-021a28b8491b">
798 <SHORT-NAME>DefaultInitValue_Double</SHORT-NAME>
799 <VALUE-SPEC>
800 <NUMERICAL-VALUE-SPECIFICATION>
801 <SHORT-LABEL>DefaultInitValue_Double</SHORT-LABEL>
802 <VALUE>0</VALUE>
803 </NUMERICAL-VALUE-SPECIFICATION>
804 </VALUE-SPEC>
805 </CONSTANT-SPECIFICATION>
806 <CONSTANT-SPECIFICATION UUID="5006437b-5b17-5fa3-f804-12599ab2177e">
807 <SHORT-NAME>DefaultInitValue_UInt16</SHORT-NAME>
808 <VALUE-SPEC>
809 <NUMERICAL-VALUE-SPECIFICATION>
810 <SHORT-LABEL>DefaultInitValue_UInt16</SHORT-LABEL>
811 <VALUE>0</VALUE>
812 </NUMERICAL-VALUE-SPECIFICATION>
813 </VALUE-SPEC>
814 </CONSTANT-SPECIFICATION>
815 </ELEMENTS>
816 </AR-PACKAGE>
817 <AR-PACKAGE UUID="54cc757e-e191-5374-e4e2-cddd9e624fd1">
818 <SHORT-NAME>DataConstrs</SHORT-NAME>
819 <ELEMENTS>
820 <DATA-CONSTR UUID="7e91eba5-e78e-5f65-e3d8-01e63d7e5417">
821 <SHORT-NAME>DC_UInt8</SHORT-NAME>
822 <DATA-CONSTR-RULES>
823 <DATA-CONSTR-RULE>

```

```
824 <INTERNAL-CONSTRS>
825 <LOWER-LIMIT INTERVAL-TYPE="CLOSED">0</LOWER-LIMIT>
826 <UPPER-LIMIT INTERVAL-TYPE="CLOSED">255</UPPER-LIMIT>
827 </INTERNAL-CONSTRS>
828 </DATA-CONSTR-RULE>
829 </DATA-CONSTR-RULES>
830 </DATA-CONSTR>
831 <DATA-CONSTR UUID="47cecee6-74ce-51ed-df40-6f6a3ff174e4">
832 <SHORT-NAME>DC_Boolean</SHORT-NAME>
833 <DATA-CONSTR-RULES>
834 <DATA-CONSTR-RULE>
835 <INTERNAL-CONSTRS>
836 <LOWER-LIMIT INTERVAL-TYPE="CLOSED">0</LOWER-LIMIT>
837 <UPPER-LIMIT INTERVAL-TYPE="CLOSED">1</UPPER-LIMIT>
838 </INTERNAL-CONSTRS>
839 </DATA-CONSTR-RULE>
840 </DATA-CONSTR-RULES>
841 </DATA-CONSTR>
842 <DATA-CONSTR UUID="15a3705a-72b5-5be8-8e75-5acff096afe4">
843 <SHORT-NAME>DC_Double</SHORT-NAME>
844 <DATA-CONSTR-RULES>
845 <DATA-CONSTR-RULE>
846 <INTERNAL-CONSTRS>
847 <LOWER-LIMIT INTERVAL-TYPE="INFINITE"/>
848 <UPPER-LIMIT INTERVAL-TYPE="INFINITE"/>
849 </INTERNAL-CONSTRS>
850 </DATA-CONSTR-RULE>
851 </DATA-CONSTR-RULES>
852 </DATA-CONSTR>
853 <DATA-CONSTR UUID="0da8cb19-2a4a-596d-3ea2-1fa242eba69d">
854 <SHORT-NAME>DC_UInt16</SHORT-NAME>
855 <DATA-CONSTR-RULES>
856 <DATA-CONSTR-RULE>
857 <INTERNAL-CONSTRS>
858 <LOWER-LIMIT INTERVAL-TYPE="CLOSED">0</LOWER-LIMIT>
859 <UPPER-LIMIT INTERVAL-TYPE="CLOSED">65535</UPPER-LIMIT>
860 </INTERNAL-CONSTRS>
861 </DATA-CONSTR-RULE>
```

```
862 </DATA-CONSTR-RULES>
863 </DATA-CONSTR>
864 <DATA-CONSTR UUID="7610c475-c999-550d-65bc-f121dff71d17">
865 <SHORT-NAME>DC_SInt32</SHORT-NAME>
866 <DATA-CONSTR-RULES>
867 <DATA-CONSTR-RULE>
868 <INTERNAL-CONSTRS>
869 <LOWER-LIMIT INTERVAL-TYPE="CLOSED">-2147483648</LOWER-LIMIT>
870 <UPPER-LIMIT INTERVAL-TYPE="CLOSED">2147483647</UPPER-LIMIT>
871 </INTERNAL-CONSTRS>
872 </DATA-CONSTR-RULE>
873 </DATA-CONSTR-RULES>
874 </DATA-CONSTR>
875 </ELEMENTS>
876 </AR-PACKAGE>
877 <AR-PACKAGE UUID="d914e838-0432-59b5-938e-6db30f1ca02c">
878 <SHORT-NAME>SwBaseTypes</SHORT-NAME>
879 <ELEMENTS>
880 <SW-BASE-TYPE UUID="f3b92e02-a26e-5036-967b-5f204c1b1587">
881 <SHORT-NAME>uint8</SHORT-NAME>
882 <CATEGORY>FIXED_LENGTH</CATEGORY>
883 <BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
884 <BASE-TYPE-ENCODING>NONE</BASE-TYPE-ENCODING>
885 <NATIVE-DECLARATION>uint8</NATIVE-DECLARATION>
886 </SW-BASE-TYPE>
887 <SW-BASE-TYPE UUID="82db31cb-3956-5eac-ee6f-bb5b2db503bf">
888 <SHORT-NAME>boolean</SHORT-NAME>
889 <CATEGORY>FIXED_LENGTH</CATEGORY>
890 <BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
891 <BASE-TYPE-ENCODING>BOOLEAN</BASE-TYPE-ENCODING>
892 <NATIVE-DECLARATION>boolean</NATIVE-DECLARATION>
893 </SW-BASE-TYPE>
894 <SW-BASE-TYPE UUID="e6110d17-9471-55b3-c8b2-cbe497c2155b">
895 <SHORT-NAME>sint32</SHORT-NAME>
896 <CATEGORY>FIXED_LENGTH</CATEGORY>
897 <BASE-TYPE-SIZE>32</BASE-TYPE-SIZE>
898 <BASE-TYPE-ENCODING>2C</BASE-TYPE-ENCODING>
899 <NATIVE-DECLARATION>sint32</NATIVE-DECLARATION>
```

```
900 </SW-BASE-TYPE>
901 <SW-BASE-TYPE UUID="903d3041-2808-56ea-9d3d-9fa5599e2a66">
902 <SHORT-NAME>uint16</SHORT-NAME>
903 <CATEGORY>FIXED_LENGTH</CATEGORY>
904 <BASE-TYPE-SIZE>16</BASE-TYPE-SIZE>
905 <BASE-TYPE-ENCODING>NONE</BASE-TYPE-ENCODING>
906 <NATIVE-DECLARATION>uint16</NATIVE-DECLARATION>
907 </SW-BASE-TYPE>
908 <SW-BASE-TYPE UUID="845df467-f0ac-57d5-7d7f-8c685231d795">
909 <SHORT-NAME>float64</SHORT-NAME>
910 <CATEGORY>FIXED_LENGTH</CATEGORY>
911 <BASE-TYPE-SIZE>64</BASE-TYPE-SIZE>
912 <BASE-TYPE-ENCODING>IEEE754</BASE-TYPE-ENCODING>
913 <NATIVE-DECLARATION>float64</NATIVE-DECLARATION>
914 </SW-BASE-TYPE>
915 </ELEMENTS>
916 </AR-PACKAGE>
917 <AR-PACKAGE UUID="5b4ebc7d-5ebc-537f-f92c-c958ad1c652f">
918 <SHORT-NAME>DataTypeMappings</SHORT-NAME>
919 <ELEMENTS>
920 <DATA-TYPE-MAPPING-SET UUID="d12839f7-61cc-5b06-e301-3a50d4a70778">
921 <SHORT-NAME>ASWCDataTypeMappingsSet</SHORT-NAME>
922 <MODE-REQUEST-TYPE-MAPS>
923 <MODE-REQUEST-TYPE-MAP>
924 <IMPLEMENTATION-DATA-TYPE-REF DEST="IMPLEMENTATION-DATA-
TYPE"/>/pkg/ASWC_dt/SInt32</IMPLEMENTATION-DATA-TYPE-REF>
925 <MODE-GROUP-REF DEST="MODE-DECLARATION-
GROUP"/>/Modes/PowerModeGroup</MODE-GROUP-REF>
926 </MODE-REQUEST-TYPE-MAP>
927 </MODE-REQUEST-TYPE-MAPS>
928 </DATA-TYPE-MAPPING-SET>
929 </ELEMENTS>
930 </AR-PACKAGE>
931 </AR-PACKAGES>
932 </AR-PACKAGE>
933 <AR-PACKAGE UUID="8c960cb7-9b4c-5653-b9d2-fb83bb9b8b1a">
934 <SHORT-NAME>imp</SHORT-NAME>
935 <ELEMENTS>
```



```

936 <SWC-IMPLEMENTATION UUID="6dbd31be-9cb6-5167-fa8a-2fafa104021c">
937 <SHORT-NAME>ASWC</SHORT-NAME>
938 <CODE-DESCRIPTORS>
939 <CODE>
940 <SHORT-NAME>Code</SHORT-NAME>
941 <ARTIFACT-DESCRIPTORS>
942 <AUTOSAR-ENGINEERING-OBJECT>
943 <SHORT-LABEL>AUTOSARDemoMdl_c</SHORT-LABEL>
944 <CATEGORY>SWSRC</CATEGORY>
945 </AUTOSAR-ENGINEERING-OBJECT>
946 <AUTOSAR-ENGINEERING-OBJECT>
947 <SHORT-LABEL>AUTOSARDemoMdl_h</SHORT-LABEL>
948 <CATEGORY>SWHDR</CATEGORY>
949 </AUTOSAR-ENGINEERING-OBJECT>
950 <AUTOSAR-ENGINEERING-OBJECT>
951 <SHORT-LABEL>OverSpeedDiagnostic_h</SHORT-LABEL>
952 <CATEGORY>SWHDR</CATEGORY>
953 </AUTOSAR-ENGINEERING-OBJECT>
954 <AUTOSAR-ENGINEERING-OBJECT>
955 <SHORT-LABEL>rtwtypes_h</SHORT-LABEL>
956 <CATEGORY>SWHDR</CATEGORY>
957 </AUTOSAR-ENGINEERING-OBJECT>
958 </ARTIFACT-DESCRIPTORS>
959 </CODE>
960 </CODE-DESCRIPTORS>
961 <PROGRAMMING-LANGUAGE>C</PROGRAMMING-LANGUAGE>
962 <RESOURCE-CONSUMPTION>
963 <SHORT-NAME>RsrcCons_AUTOSARDemoMdl</SHORT-NAME>
964 </RESOURCE-CONSUMPTION>
965 <SW-VERSION>1.273.1</SW-VERSION>
966 <USED-CODE-GENERATOR>Embedded Coder 6.10 (R2016a) 10-Feb-2016</USED-
CODE-GENERATOR>
967 <VENDOR-ID>0</VENDOR-ID>
968 <BEHAVIOR-REF DEST="SWC-INTERNAL-
BEHAVIOR">/pkg/swc/ASWC/ASWC</BEHAVIOR-REF>
969 </SWC-IMPLEMENTATION>
970 </ELEMENTS>
971 </AR-PACKAGE>

```

```

972 </AR-PACKAGES>
973 </AR-PACKAGE>
974 <AR-PACKAGE UUID="044a89d0-542f-4868-aa54-ef0465d9b394">
975 <SHORT-NAME>Modes</SHORT-NAME>
976 <ELEMENTS>
977 <MODE-DECLARATION-GROUP UUID="1fb7be08-aabc-5d78-574f-6a1b2206e067">
978 <SHORT-NAME>PowerModeGroup</SHORT-NAME>
979 <DESC>
980 <L-2 L="FOR-ALL">Vehicle Power Modes</L-2>
981 </DESC>
982 <INITIAL-MODE-REF DEST="MODE-
DECLARATION"/>Modes/PowerModeGroup/CRANK</INITIAL-MODE-REF>
983 <MODE-DECLARATIONS>
984 <MODE-DECLARATION UUID="6b352cec-b042-52f9-0b88-5a54455999e8">
985 <SHORT-NAME>ACC</SHORT-NAME>
986 <VALUE>0</VALUE>
987 </MODE-DECLARATION>
988 <MODE-DECLARATION UUID="773b3640-33e4-5397-7876-63a5f2371930">
989 <SHORT-NAME>CRANK</SHORT-NAME>
990 <VALUE>1</VALUE>
991 </MODE-DECLARATION>
992 <MODE-DECLARATION UUID="584c1345-e233-5fd0-4ead-f6f60219c94e">
993 <SHORT-NAME>RUN</SHORT-NAME>
994 <VALUE>2</VALUE>
995 </MODE-DECLARATION>
996 <MODE-DECLARATION UUID="ac7a2720-bf0e-5084-8c2e-201d1ea16d23">
997 <SHORT-NAME>OFF</SHORT-NAME>
998 <VALUE>3</VALUE>
999 </MODE-DECLARATION>
100 <MODE-DECLARATION UUID="f48b2426-1878-5203-3e55-6c02e4989061">
0
100 <SHORT-NAME>SLEEP</SHORT-NAME>
1
100 <VALUE>4</VALUE>
2
100 </MODE-DECLARATION>
3
100 </MODE-DECLARATIONS>
4
100 </MODE-DECLARATION-GROUP>

```

```
5
100 <MODE-SWITCH-INTERFACE UUID="93d5ca3a-1d9a-4cb7-9bea-484115c3ea7e">
6
100 <SHORT-NAME>PowerModelInterface</SHORT-NAME>
7
100 <IS-SERVICE>>false</IS-SERVICE>
8
100 <MODE-GROUP UUID="b6b65637-d551-479d-905f-980feaddc6b8">
9
101 <SHORT-NAME>PowerModeData</SHORT-NAME>
0
101 <SW-CALIBRATION-ACCESS>READ-WRITE</SW-CALIBRATION-ACCESS>
1
101 <TYPE-TREF                                DEST="MODE-DECLARATION-
2   GROUP">/Modes/PowerModeGroup</TYPE-TREF>
101 </MODE-GROUP>
3
101 </MODE-SWITCH-INTERFACE>
4
101 </ELEMENTS>
5
101 </AR-PACKAGE>
6
101 </AR-PACKAGES>
7
101 </AUTOSAR>
8
```