ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχεδίαση και Ανάπτυξη σύνθετων εφαρμογών και διασυνδέσεων συστημάτων με βάση το Embed λειτουργικό σύστημα ενσωματωμένων συστημάτων

Βασίλης Καραγιάννης

Επιβλέπων καθηγητές : **Επίκουρος Καθηγητής Χρήστος Αντωνόπουλος**

Πάτρα, 2022

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Πάτρα, 30/6/2023

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Χρήστος Αντωνόπουλος, Επίκουρος καθηγητής

2. Νικόλαος Βώρος, Καθηγητής

3. Νικόλαος Πετρέλης, Καθηγητής

ΠΡΟΛΟΓΟΣ

Τα τελευταία χρόνια παρατηρούμε μια μεγάλη εξέλιξη στον τομέα της τεχνολογίας και ιδιαίτερα στο κομμάτι των ενσωματωμένων συστημάτων τα οποία έχουν συμβάλει σε πολύ μεγάλό βαθμό στην καθημερινή μας ζωή. Ως ενσωματωμένο σύστημα μπορεί να οριστεί ένα σύστημα – περιορισμένου φυσικού μεγέθους – ειδικού σκοπού το οποίο επιτελεί μία συγκεκριμένη κριμένη και προκαθορισμένη εργασία.

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι η μελέτη ενσωματωμένων συστημάτων, των περιφερειακών τους και των λειτουργικών συστημάτων τους. Με βάση αυτό θα αναπτυχθούν σύνθετες εφαρμογές εκμεταλλευόμενη και κατανοώντας σε βάθος κυρίαρχο προσομοιωτή. Στο τέλος θα γινεί και διασύνδεση μεταξύ συστημάτων.

Υπεύθυνος κατά την εκπόνηση της διπλωματικής εργασίας ήταν ο Καθηγητής κ. Χρήστος Αντωνόπουλος. Επίσης θα ήθελα να ευχαριστήσω και τον υποψήφιο διδάκτορα τον κ. Χρήστο Παναγιώτου ο οποίος σύμβαλλε και εκείνος για την εκπόνηση την διπλωματικής .

ΠΕΡΙΛΗΨΗ

Η εργασία αρχικά περιλαμβάνει το εισαγωγικό κεφάλαιο, όπου γίνεται αναφορά στα ενσωματωμένα συστήματα ως προς το θεωρητικό κομμάτι. Παράλληλα δίνεται έμφαση και στα λειτουργικά συστήματα που και εκεί θα αναφέρουμε ως προς τον ορισμό-λειτουργία τους. Στη συνέχεια, θα προχωρήσουμε στο επόμενο κεφάλαιο το οποίο έχει να κάνει με την αρχιτεκτονική επεξεργαστών. Επιπλέον, θα δώσουμε ιδιαίτερη σημασία στο λειτουργικό σύστημα Embed OS,όπου και θα δείξουμε εκτελέσιμο κώδικα πάνω σε αυτό. Επίσης θα πραγματοποιηθούν μερικά demo βασισμένα στο Embed OS.Τέλος, θα παρουσιαστεί αναλυτικά η διαδικασία-εργαλεία εγκατάστασης για να πραγματοποιηθούν τα συγκεκριμένα demo.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1°: Ορισμός ενός ενσωματωμένου συστήματος

1.1 Εισαγωγή

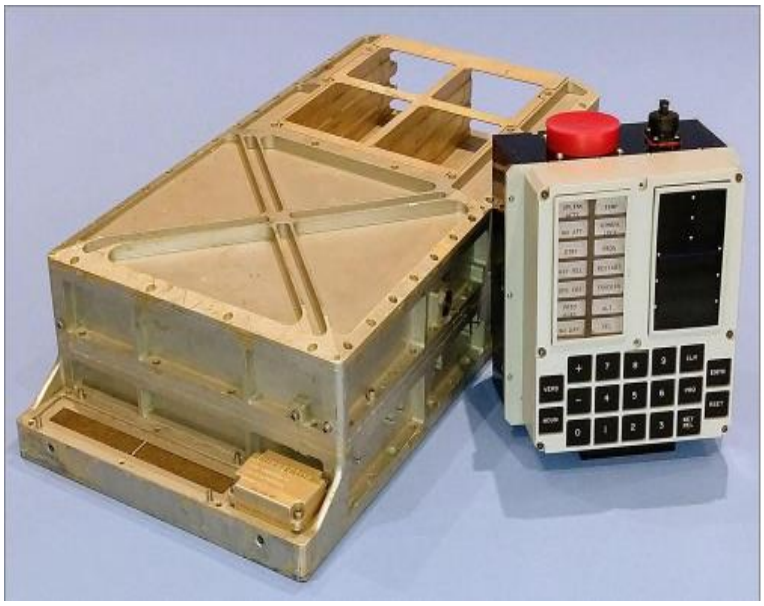
Τα τελευταία χρόνια, λόγω της αυξημένης ζήτησης σε υπολογιστική ισχύ σε συνδυασμό με την απαίτηση για μεταφερσιμότητα (portability) παρατηρείται μια μεγάλη αύξηση του πεδίου που βρίσκουν εφαρμογή τα Ενσωματωμένα Συστήματα. Ως Ενσωματωμένο Σύστημα (ΕΣ) μπορεί να οριστεί μία οποιαδήποτε συσκευή η οποία περιλαμβάνει έναν προγραμματιζόμενο υπολογιστή, ο οποίος δεν είναι υπολογιστής γενικού σκοπού. Πρόκειται δηλαδή για συστήματα τα οποία απαιτούν κάποιον έλεγχο, λόγου χάρη ένας οικιακός αυτοματισμός, το οποίο καλείται να εκτελέσει ένας υπολογιστής που έχει ενσωματωθεί σε αυτά. Το ενσωματωμένο σύστημα είναι προγραμματιζόμενο ή μη-προγραμματιζόμενο, ανάλογα με την εφαρμογή για την οποία σχεδιάζεται. Όλα τα τμήματα ενώνονται και λειτουργούν μαζί σύμφωνα με κάποιο εκτελέσιμο πρόγραμμα.

Αυτό που διαφοροποιεί τα ενσωματωμένα συστήματα από τα υπόλοιπα πληροφοριακά συστήματα είναι η ένωση τους με άλλα συστήματα δεν περιορίζεται σε μία απλή συνδεσμολογία αλλά στην ενσωμάτωσή τους μέσα σε ένα γενικότερο/μεγαλύτερο σύστημα, το οποίο πιθανόν να "αποκρύπτει" και την ύπαρξη του ΕΣ.

1.2 Ιστορική Αναδρομή των ΕΣ

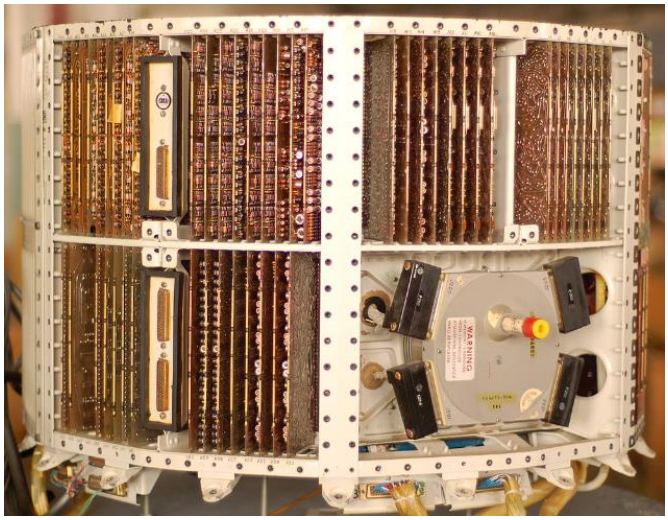
Η ραγδαία ανάπτυξη των υπολογιστικών συστημάτων έχει επηρεάσει κυρίως την κατηγορία των "μικρό-υπολογιστών". Αυτό έχει ως αποτέλεσμα οι διάφορες υποκατηγορίες όπως οι προσωπικοί υπολογιστές (PC) και τα ενσωματωμένα συστήματα να τείνουν όλο και περισσότερο να συγχωνευτούν μμεταξύ τους. Έτσι, όσο κυλάει ο χρόνος γίνεται όλο και πιο δύσκολο να ξεχωρίσει κανείς τους προσωπικούς υπολογιστές από τα ενσωματωμένα συστήματα.

Το πρώτο σύγχρονο, ενσωματωμένο υπολογιστικό σύστημα πραγματικού χρόνου ήταν ο υπολογιστής καθοδήγησης apollo, που αναπτύχθηκε το 1960 από τον Charles Stark στο Ινστιτούτο Τεχνολογίας της Μασαχουσέτης για το πρόγραμμα apollo. Ο υπολογιστής καθοδήγησης apollo σχεδιάστηκε για να συλλέγει αυτόματα δεδομένα και να παρέχει κρίσιμους υπολογισμούς για την αποστολή της μονάδας διοίκησης apollo και της σεληνιακής μονάδας. Επιπλέον, χρησιμοποιήθηκε για να βοηθήσει στην πλοήγηση προς το φεγγάρι σε εννέα εκτοξεύσεις. Έξι από αυτές τις εκτοξεύσεις οδήγησαν σε επιτυχή προσγείωση στην επιφάνεια της σελήνης. Κάθε σεληνιακή αποστολή διέθετε δύο επιπλέον υπολογιστές για τον έλεγχο του σκάφους. Ο ψηφιακός υπολογιστής οχήματος εκτόξευσης (LVDC) και το σύστημα καθοδήγησης ματαίωσης (AGS) επέτρεπαν στους αστροναύτες να απογειωθούν από το φεγγάρι και να συναντηθούν με τη μονάδα διακυβέρνησης. Ωστόσο, δεν μπορούσε να προσγειωθεί χρησιμοποιώντας τους υπολογιστές. Την ίδια δεκαετία, δημιουργήθηκε ο υπολογιστής D-17B ως μέρος του συστήματος καθοδήγησης πυραύλων Minuteman.



Εικόνα 1: Apollo Guidance Computer (AGC),1960

Για τον πύραυλο Minuteman I, το πλήρες σύστημα καθοδήγησης πυραύλων αποτελούνταν από έναν υπολογιστή D-17B, τροφοδοτικά και μια σταθερή πλατφόρμα. Ο υπολογιστής D-17B περιείχε 6282 διόδους, 1521 τρανζίστορ, 1116 πυκνωτές και 504 αντιστάσεις. Πλήρως συναρμολογημένο το σύστημα καθοδήγησης ζύγιζε περίπου 62 κιλά. Καθένα από αυτά τα επιμέρους εξαρτήματα ήταν τοποθετημένο σε διπλές χάλκινες, επιχρυσωμένες, πλαστικοποιημένες πλακέτες κυκλωμάτων από ίνες γυαλιού. Συνολικά 75 από αυτές τις πλακέτες κυκλωμάτων αποτελούν τον υπολογιστή D-17B και κάθε μία ήταν επικαλυμμένη με μια εύκαμπτη ένωση πολυουρεθάνης σχεδιασμένη για προστασία από την υγρασία και τους κραδασμούς. Αυτό το οπλικό σύστημα είχε ακραίες απαιτήσεις αξιοπιστίας και ανθεκτικότητας.



Εικόνα 2: Υπολογιστής συστήματος καθοδήγησης πυραύλων Autonetics D 17B

1.3 Χαρακτηριστικά

Σε αντίθεση με τα γενικά συστήματα υπολογιστών, τα ενσωματωμένα συστήματα λειτουργούν μόνο για μια συγκεκριμένη λειτουργία με χρονικά περιορισμένο τρόπο. Για παράδειγμα, ένα πλυντήριο ρούχων δεν μπορεί να κάνει πολλές εργασίες όπως ένας φορητός υπολογιστής. Από αυτή την άποψη, εδώ είναι μερικά μοναδικά χαρακτηριστικά ενός ενσωματωμένου συστήματος.

- Εκλεπτυσμένη λειτουργικότητα

Η λειτουργικότητα κανενός ενσωματωμένου συστήματος δεν είναι βέβαιο ότι είναι ίδια. Η λειτουργικότητα ενός πλυντηρίου ρούχων είναι διαφορετική από εκείνη ενός φούρνου μικροκυμάτων. Ωστόσο, η λειτουργικότητα ενός φορητού υπολογιστή και ενός επιτραπέζιου υπολογιστή είναι σχεδόν η ίδια.

- Λειτουργία σε πραγματικό χρόνο

Το λογισμικό προγραμματίζει το υλικό ώστε να λειτουργεί με τρόπο που να είναι χρονικά περιορισμένος. Θα μπορούσε επίσης να έχει δύο τρόπους λειτουργίας: Σκληρή και μαλακή. Ο πρώτος τρόπος υποδηλώνει ότι η εργασία πρέπει να ολοκληρωθεί εντός του καθορισμένου χρόνου (π.χ. ρολόι), αλλά στον μαλακό τρόπο λειτουργίας, το σύστημα θα μπορούσε να χρησιμοποιήσει επιπλέον χρόνο πέρα από τον καθορισμένο χρόνο (π.χ. μικροκύματα).

- Χαμηλό κόστος κατασκευής

Καθώς ο σχεδιασμός ενός ενσωματωμένου συστήματος στοχεύει σε οποιαδήποτε συγκεκριμένη εφαρμογή, συνεπάγεται μικρότερο κόστος κατασκευής σε σύγκριση με ένα ευέλικτο γενικό υπολογιστικό σύστημα. Κατά συνέπεια, τα ενσωματωμένα συστήματα απαιτούν επίσης λιγότερη ισχύ για την εκτέλεση λειτουργιών.

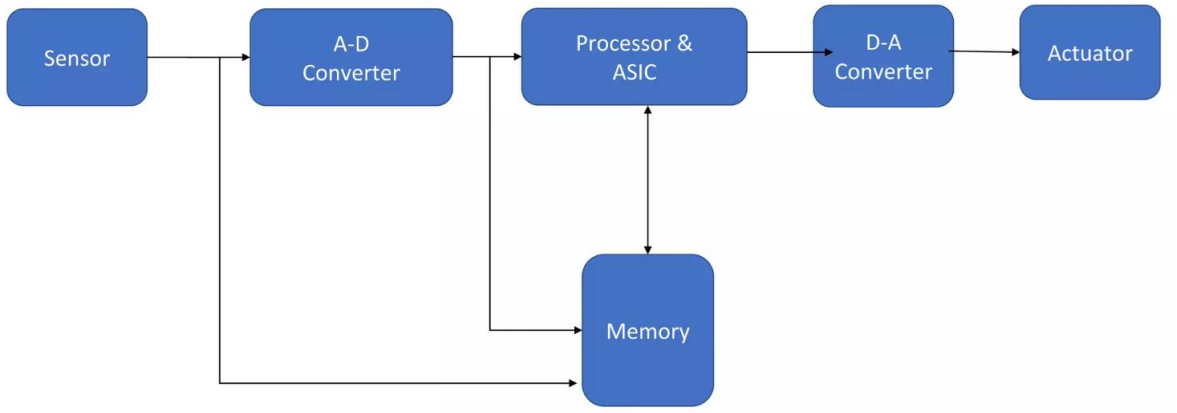
- Επεξεργαστής και μνήμη

Ανάλογα με τον τύπο, οι απαιτήσεις επεξεργαστή και μνήμης ενδέχεται να διαφέρουν. Για παράδειγμα, τα μικρά ενσωματωμένα συστήματα θα απαιτούσαν λιγότερη μνήμη, αλλά τα εξελιγμένα συστήματα απαιτούν περισσότερη μνήμη και λειτουργούν σε πολυπύρηνους επεξεργαστές.

- Στενός περιορισμός σχεδιασμού

Υπάρχουν πολλοί σχεδιαστικοί περιορισμοί που πρέπει να ληφθούν υπόψη γύρω από το κόστος, τις επιδόσεις, το μέγεθος και την ισχύ ενός ενσωματωμένου συστήματος για να επιτευχθεί η απόλυτη απόδοσή του. Αυτοί οι σχεδιαστικοί παράγοντες διατηρούνται στο ελάχιστο για να δικαιολογούν την απλή λειτουργία τους.

1.4 Δομή ενός ΕΣ



Ένας αισθητήρας αποθηκεύει τη μετρούμενη ποσότητα στη μνήμη

Μετατρέπει το αναλογικό σήμα που στέλνει ο αισθητήρας σε ψηφιακό σήμα

Επεξεργάζεται τα δεδομένα για τη μέτρηση της εξόδου και τα αποθηκεύει στη μνήμη

Μετατρέπει τα ψηφιακά δεδομένα που τροφοδοτούνται από τον επεξεργαστή σε αναλογικά δεδομένα

Συγκρίνει την έξοδο που δίνει ο μετατροπέας D-A με την πραγματική (αναμενόμενη) έξοδο που είναι αποθηκευμένη σε αυτόν και αποθηκεύει την εγκεκριμένη έξοδο

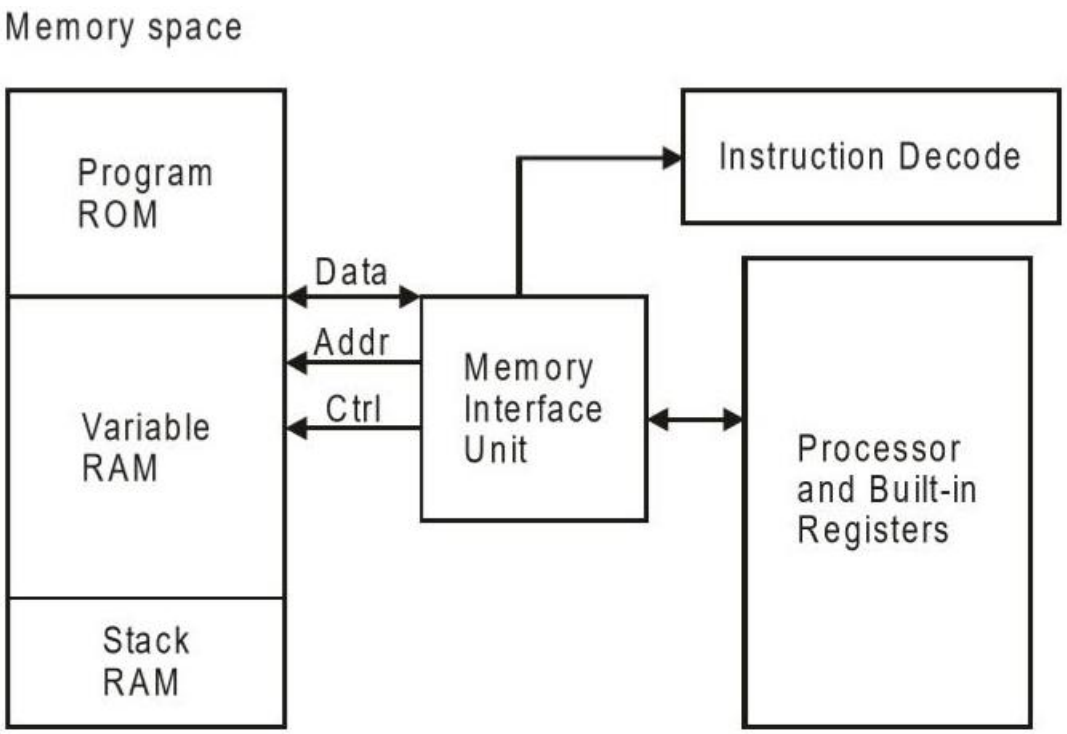
1.5 Αρχιτεκτονικές

1.5.1 αρχιτεκτονικές VON NEUMANN και HARVARD

Το τυπικό ενσωματωμένο σύστημα αποτελείται κυρίως από δύο μέρη, δηλαδή, ενσωματωμένο υλικό και ενσωματωμένο λογισμικό. Το ενσωματωμένο σκληρό λογισμικό βασίζεται σε μικροεπεξεργαστές και μικροελεκτές, περιλαμβάνει επίσης μνήμη, δίαυλο, είσοδο/έξοδο, ελεγκτή, ενώ το ενσωματωμένο λογισμικό περιλαμβάνει ενσωματωμένα λειτουργικά συστήματα, διάφορες εφαρμογές και προγράμματα οδήγησης συσκευών. Βασικά αυτοί οι δύο τύποι αρχιτεκτονικής, δηλαδή η αρχιτεκτονική Harvard και η αρχιτεκτονική Von Neumann, χρησιμοποιούνται στα ενσωματωμένα συστήματα.

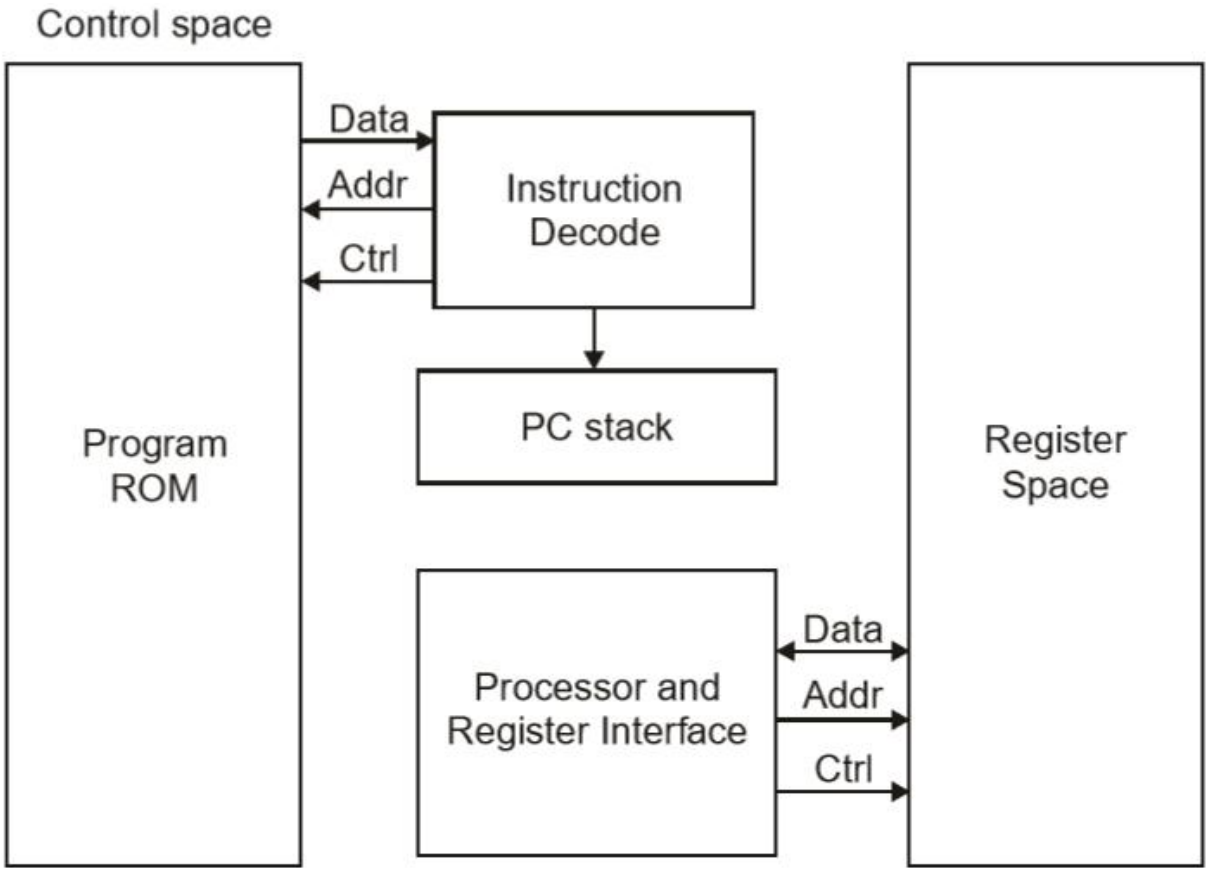
VON NEUMANN

Η αρχιτεκτονική Von Neumann προτάθηκε για πρώτη φορά από τον επιστήμονα υπολογιστών John von Neumann. Σε αυτή την αρχιτεκτονική, ένα μονοπάτι δεδομένων ή δίαυλος υπάρχει τόσο για τις εντολές όσο και για τα δεδομένα. Ως αποτέλεσμα, η CPU εκτελεί μία λειτουργία κάθε φορά. Είτε ανακτά μια εντολή από τη μνήμη, είτε εκτελεί λειτουργία ανάγνωσης/εγγραφής σε δεδομένα. Έτσι, δεν μπορεί να γίνει ταυτόχρονα μια ανάκληση εντολής και μια λειτουργία δεδομένων, που μοιράζονται έναν κοινό δίαυλο. Επιπλέον, η αρχιτεκτονική υποστηρίζει απλό υλικό. Επιτρέπει τη χρήση μιας μόνο, διαδοχικής μνήμης. Οι σημερινές ταχύτητες επεξεργασίας ξεπερνούν κατά πολύ τους χρόνους πρόσβασης στη μνήμη και χρησιμοποιούμε μια πολύ γρήγορη αλλά μικρή ποσότητα μνήμης (cache) τοπικά στον επεξεργαστή.



HARVARD

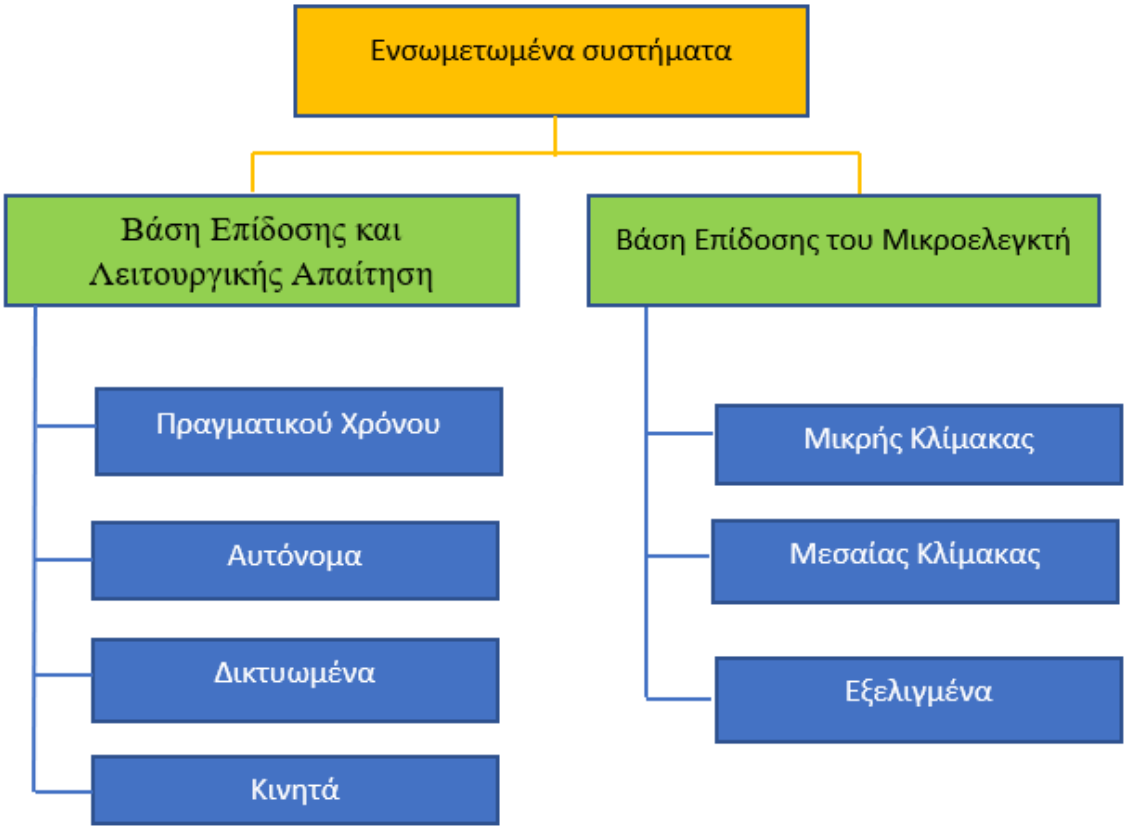
Η αρχιτεκτονική Harvard προσφέρει ξεχωριστούς διαύλους αποθήκευσης και σήματος για τις εντολές και τα δεδομένα. Αυτή η αρχιτεκτονική έχει την αποθήκευση δεδομένων εξ ολοκλήρου εντός της CPU και δεν υπάρχει πρόσβαση στην αποθήκευση εντολών ως δεδομένα. Οι υπολογιστές διαθέτουν ξεχωριστές περιοχές μνήμης για τις εντολές του προγράμματος και τα δεδομένα χρησιμοποιώντας εσωτερικούς διαύλους δεδομένων, επιτρέποντας την ταυτόχρονη πρόσβαση τόσο στις εντολές όσο και στα δεδομένα.



Von Neumann αρχιτεκτονική	Harvard αρχιτεκτονική
Ενιαία μνήμη που μοιράζονται τόσο ο κώδικας όσο και τα δεδομένα.	Ξεχωριστές μνήμες για τον κώδικα και τα δεδομένα.
Ο επεξεργαστής πρέπει να αντλεί τον κώδικα σε ξεχωριστό κύκλο ρολογιού και τα δεδομένα σε άλλο κύκλο ρολογιού. Επομένως, απαιτούνται δύο κύκλοι ρολογιού.	Ένας κύκλος ρολογιού είναι επαρκής, καθώς χρησιμοποιούνται ξεχωριστοί δίαυλοι για την πρόσβαση στον κώδικα και τα δεδομένα.
Υψηλότερη ταχύτητα, άρα λιγότερη χρονοβόρα.	Χαμηλότερη ταχύτητα, επομένως πιο χρονοβόρα.
Απλός σχεδιασμός.	Πολύπλοκος σχεδιασμός.

1.6 Κατηγορίες ΕΣ

Μπορούμε να κατηγοριοποιήσουμε τα ενσωματωμένα συστήματα με βάση τις επιδόσεις και τις λειτουργικές απαιτήσεις και με βάση τις επιδόσεις του μικροελεκτή. Ο μικροελεκτής είναι σημαντικός για ένα ενσωματωμένο σύστημα, καθώς η λειτουργία του υλικού εξαρτάται πλήρως από τον μικροελεκτή. Φροντίζει για την επικοινωνία και τους επεξεργαστές που απαιτούνται για το σύστημα. Οι απαιτήσεις του χρήστη εξαρτώνται από τη λειτουργία του ενσωματωμένου συστήματος.



Βάση επίδοσης και λειτουργικής απαιτήσεις

- Πραγματικού Χρόνου
  Ένα ενσωματωμένο σύστημα πραγματικού χρόνου παρέχει έξοδο εντός ενός καθορισμένου συγκεκριμένου χρόνου. Δηλαδή, τα ενσωματωμένα συστήματα πραγματικού χρόνου σχεδιάζονται και δημιουργούνται για να εκτελούν κάποια συγκεκριμένη εργασία σε προκαθορισμένο χρόνο.
- Αυτόνομα
  Τα αυτόνομα ενσωματωμένα συστήματα είναι εκείνα που μπορούν να λειτουργήσουν μόνα τους, δηλαδή είναι αυτάρκη και δεν εξαρτώνται από ένα κεντρικό σύστημα. Τα αυτόνομα ενσωματωμένα συστήματα κατασκευάζονται με τέτοιο τρόπο ώστε να λαμβάνεται μια είσοδος, να επεξεργάζεται και στη συνέχεια να παράγεται η επιθυμητή έξοδος.

- Δικτυωμένα
  Τα δικτυωμένα ενσωματωμένα συστήματα εξαρτώνται από το συνδεδεμένο δίκτυο για την εκτέλεση των καθηκόντων που τους έχουν ανατεθεί. Τα συστήματα αυτά αποτελούνται από στοιχεία όπως αισθητήρες, ελεγκτές κ.λπ. τα οποία είναι διασυνδεδεμένα. Πολλά από αυτά τα συστήματα βασίζονται σε επεξεργαστές γενικού σκοπού.

- Κινητά
  Τα κινητά ενσωματωμένα συστήματα είναι αυτά που έχουν μικρό μέγεθος και μπορούν να χρησιμοποιηθούν σε μικρότερες συσκευές. Χρησιμοποιούνται σε κινητά τηλέφωνα και ψηφιακές φωτογραφικές μηχανές λόγω του μικρού τους μεγέθους. Συχνά έχουν περιορισμούς μνήμης και δεν διαθέτουν καλή διεπαφή χρήστη.

Βάση επίδοσης του μικροελεκτή

- Μικρής κλίμακας
  Το ενσωματωμένο σύστημα μικρής κλίμακας συνήθως σχεδιάζεται και δημιουργείται με τη χρήση ενός μικροελεκτή 8 bit. Αυτός ο μικροελεκτής μπορεί να ενεργοποιηθεί με μπαταρία.

- Μεσαίας κλίμακας
  Το Ενσωματωμένο σύστημα μεσαίας κλίμακας χρησιμοποιεί έναν ενιαίο μικροελεκτή 16 ή 32 bit ή πολλούς μικροελεκτής συνδεδεμένους μεταξύ τους. Αυτά τα συστήματα έχουν πολλές πολυπλοκότητες υλικού καθώς και λογισμικού, και ως εκ τούτου δεν προτιμώνται από πολλούς.

- Εξελιγμένα
  Τα εξελιγμένα ενσωματωμένα συστήματα συχνά λειτουργούν με πολλαπλούς αλγορίθμους, γεγονός που οδηγεί σε πολυπλοκότητα τόσο στο υλικό όσο και στο λογισμικό. Συχνά χρειάζονται έναν επεξεργαστή που είναι διαμορφώσιμος και μια λογική συστοιχία που μπορεί να προγραμματιστεί.

1.7 Προκλήσεις στον σχεδιασμό

Το ενσωματωμένο λογισμικό αποτελεί πάντα συστατικό ενός ευρύτερου συστήματος, για παράδειγμα, ενός ψηφιακού ρολογιού, ενός smartphone, ενός οχήματος ή ενός αυτοματοποιημένου βιομηχανικού εξοπλισμού. Τέτοιες ενσωματωμένες λύσεις πρέπει να έχουν απόκριση σε πραγματικό χρόνο υπό όλες τις συνθήκες εντός του χρόνου που ορίζει ο σχεδιασμός και να λειτουργούν υπό συνθήκες περιορισμένης μνήμης, επεξεργαστικής ισχύος και παροχής ενέργειας. Επιπλέον, το ενσωματωμένο λογισμικό πρέπει να είναι απρόσβλητο από τις αλλαγές στο περιβάλλον λειτουργίας του - οι επεξεργαστές, οι αισθητήρες και τα εξαρτήματα υλικού μπορεί να αλλάζουν με την πάροδο του χρόνου

Σταθερότητα

Η σταθερότητα είναι υψίστης σημασίας. Η μη αναμενόμενη συμπεριφορά ενός ενσωματωμένου λογισμικού είναι ανεπίτρεπτη και εγκυμονεί σοβαρούς κινδύνους. Οι τελικοί χρήστες απαιτούν από το ενσωματωμένο λογισμικό να έχει ομοιόμορφη συμπεριφορά υπό όλες τις συνθήκες και να είναι σε θέση να λειτουργεί διαρκώς χωρίς σέρβις.

Ασφάλεια

Η ασφάλεια είναι ένα ιδιαίτερο χαρακτηριστικό του ενσωματωμένου λογισμικού λόγω της πρωταρχικής εφαρμογής τους που σχετίζεται με τη λειτουργία διάσωσης ζωών σε κρίσιμα περιβάλλοντα. Ο κύκλος ζωής ανάπτυξης λογισμικού (SDLC) για το ενσωματωμένο λογισμικό χαρακτηρίζεται από πιο αυστηρές απαιτήσεις και περιορισμούς όσον αφορά την ποιότητα, τις δοκιμές και την τεχνογνωσία των μηχανικών. Αυτός είναι ο λόγος για τον οποίο η ασφάλεια αποτελεί μία από τις μεγαλύτερες προκλήσεις στην ανάπτυξη ενσωματωμένων συστημάτων.

Περιορισμοί σχεδιασμού

Οι προκλήσεις στη σχεδίαση του ενσωματωμένου λογισμικού είναι πάντα οι ίδιες περιοριστικές απαιτήσεις εδώ και δεκαετίες:

-Μικρός παράγοντας μορφής,

-Χαμηλή ενέργεια,

-Μακροχρόνια σταθερή απόδοση χωρίς συντήρηση.

Ανάλογα με τις εφαρμογές στο IoT και σε άλλους τομείς, υπάρχει αυξανόμενη ζήτηση για την κατασκευή επεξεργαστών που είναι κλιμακούμενοι ή όχι μεγάλοι σε μέγεθος με φθηνή και εξαιρετικά χαμηλή κατανάλωση ενέργειας για μέγιστη απόδοση χωρίς συντήρηση.

Συμβατότητα και Ακεραιότητα

Οι περισσότερες από τις εφαρμογές για κινητά που έχουμε στο App Store κατασκευάστηκαν από εταιρείες όταν λίγο πολύ ήταν μικρότερες από 3 ετών. Η ακεραιότητα και η ασφάλεια είναι πολύ ουσιαστικά και αδιαχώριστα μέρη και δεν υπάρχουν το ένα χωρίς το άλλο όταν πρόκειται για το IoT. Οι συσκευές IoT πρέπει να προστατεύονται κατάλληλα από διάφορες κακόβουλες επιθέσεις στον κυβερνοχώρο και επίσης πρέπει να εφαρμόζεται ασφάλεια σε κάθε επίπεδο ανάπτυξης λογισμικού.

Ρυθμός αλλαγής

Στο πρόσφατο παρελθόν, οι τεχνολογίες που χρησιμοποιούνται για την ανάπτυξη ενσωματωμένων συστημάτων ήταν σχεδόν οι ίδιες με κάποιο νέο επεξεργαστή υψηλότερης χωρητικότητας που θα κυκλοφορούσε μία φορά το χρόνο. Τότε τα πράγματα άρχισαν τελικά να επιταχύνονται. Τα τελευταία 5 χρόνια, γινόμαστε μάρτυρες της γρήγορης ανάπτυξης αναδυόμενων τεχνολογιών, συμπεριλαμβανομένης της τεχνητής νοημοσύνης. Έτσι, αυτό δημιουργεί πρόβλημα στους προγραμματιστές, επειδή οι διαθέσιμες τεχνολογίες αλλάζουν πιο γρήγορα από ό,τι μπορούν να τις χειριστούν. Αυτό λοιπόν καθιστά αυτή τη δυναμική αλλαγή πρόκληση για τα ενσωματωμένα συστήματα.

1.8 Μεθοδολογίες στον σχεδιασμό

Τα περισσότερα πραγματικά σύγχρονα σχέδια ενσωματωμένων συστημάτων είναι εγγενώς πολύπλοκα, δεδομένου ότι οι λειτουργικές προδιαγραφές τους είναι πλούσιες και πρέπει να ακούν σε πολλαπλές άλλες απαιτήσεις σχετικά με το κόστος, τις επιδόσεις κ.ο.κ. Ως αποτέλεσμα, χρειαζόμαστε μεθοδολογίες και πλαίσια που θα μας βοηθήσουν να καθοδηγήσουμε τις αποφάσεις μας κατά τη σχεδίαση μεγάλων συστημάτων. Ο προφανής στόχος μιας διαδικασίας σχεδιασμού είναι η δημιουργία ενός προϊόντος που κάνει κάτι χρήσιμο. Οι τυπικές προδιαγραφές για ένα προϊόν θα περιλαμβάνουν λειτουργικότητα , κόστος κατασκευής , απόδοση , κατανάλωση ενέργειας ή άλλες ιδιότητες. Φυσικά, μια διαδικασία σχεδιασμού έχει αρκετούς σημαντικούς στόχους πέρα από τη λειτουργία, την απόδοση και την ισχύ. Τρεις από αυτούς τους στόχους συνοψίζονται παρακάτω.

Χρόνος διάθεσης στην αγορά: Οι πελάτες θέλουν πάντα νέα χαρακτηριστικά. Το προϊόν που βγαίνει πρώτο μπορεί να κερδίσει την αγορά, καθορίζοντας ακόμη και τις προτιμήσεις των πελατών για τις μελλοντικές γενιές του προϊόντος. Σε ορισμένες κατηγορίες, ο ανταγωνισμός αφορά το ημερολόγιο και όχι μόνο τους ανταγωνιστές.

Κόστος σχεδιασμού: Πολλά καταναλωτικά προϊόντα είναι πολύ ευαίσθητα στο κόστος. Οι βιομηχανικοί αγοραστές ενδιαφέρονται επίσης όλο και περισσότερο για το κόστος. Το κόστος σχεδιασμού του συστήματος διαφέρει από το κόστος κατασκευής - το κόστος των μισθών των

μηχανικών, των υπολογιστών που χρησιμοποιούνται στο σχεδιασμό κ.λπ. πρέπει να κατανεμηθεί στις μονάδες που πωλούνται. Σε ορισμένες περιπτώσεις, μπορεί να κατασκευαστεί μόνο ένα ή λίγα αντίγραφα ενός ενσωματωμένου συστήματος, οπότε το κόστος σχεδιασμού μπορεί να κυριαρχήσει στο κόστος κατασκευής.

Ποιότητα: Οι πελάτες δεν θέλουν μόνο τα προϊόντα τους γρήγορα και φθηνά, αλλά και να είναι σωστά. Μια μεθοδολογία σχεδιασμού που παράγει άθλια προϊόντα θα εκδιωχθεί σύντομα από την αγορά. Η ορθότητα, η αξιοπιστία και η ευχρηστία πρέπει να αντιμετωπίζονται ρητά από την αρχή της εργασίας σχεδιασμού, ώστε στο τέλος να προκύψει ένα προϊόν υψηλής ποιότητας. Μια καλή μεθοδολογία είναι ζωτικής σημασίας για τη δημιουργία συστημάτων που λειτουργούν σωστά. Η παράδοση προβληματικών συστημάτων στους πελάτες προκαλεί πάντα δυσαρέσκεια. Αλλά σε ορισμένες εφαρμογές, όπως τα ιατρικά συστήματα και τα συστήματα αυτοκινήτων, τα σφάλματα δημιουργούν σοβαρά προβλήματα ασφάλειας που μπορούν να θέσουν σε κίνδυνο τη ζωή των χρηστών.

1.9 Παραδείγματα-Εφαρμογές

Τα ενσωματωμένα συστήματα αρχικά, είχαν εμφανιστεί ως συσκευές ή προϊόντα χαμηλών επιδόσεων, λόγου χάρη ρολόγια και ηλεκτρονικοί υπολογιστές τσέπης. Ωστόσο σήμερα, στις τεχνολογικά αναπτυγμένες κοινωνίες, υπάρχουν πολλά πράγματα με ενσωματωμένα συστήματα που ενσωματώνονται στο Διαδίκτυο των Πραγμάτων (IoT), καθώς και σε συσκευές μηχανής προς μηχανή (M2M). Εξαιρετικά ευέλικτα και προσαρμόσιμα, τα ενσωματωμένα συστήματα βρίσκονται σήμερα σε όλες τις έξυπνες συσκευές. Είναι δύσκολο να βρεθεί έστω και ένα τμήμα της σύγχρονης ζωής που να μην περιλαμβάνει αυτή την τεχνολογία. Ακολουθούν μερικά από τα παραδείγματα εφαρμογών ενσωματωμένων συστημάτων στην πραγματική ζωή.

Συστήματα κεντρικής θέρμανσης

Τα συστήματα κεντρικής θέρμανσης μετατρέπουν τη χημική ενέργεια σε θερμική ενέργεια σε ένα δωμάτιο κλιβάνου και μεταφέρουν την ενέργεια αυτή σε θερμότητα, η οποία στη συνέχεια παρέχεται σε πολυάριθμους χώρους εντός του κτιρίου. Είναι σημαντικό για τα συστήματα αυτά να διαθέτουν θερμοστάτες ελέγχου για τη ρύθμιση της θερμοκρασίας, η οποία επιτυγχάνεται από ένα ενσωματωμένο σύστημα. Παραδείγματα ενσωματωμένου συστήματος στην κεντρική θέρμανση μπορεί να βρεθούν σε μια σειρά από κατασκευές που απαιτούν έλεγχο της θερμοκρασίας, τόσο για λόγους άνεσης όσο και για τη διαχείριση ευαίσθητων στη θερμοκρασία αγαθών.

Συστήματα GPS

Το GPS είναι ένα σύστημα πλοήγησης που χρησιμοποιεί δορυφόρους και δέκτες για το συγχρονισμό δεδομένων σχετικά με τη θέση, το χρόνο και την ταχύτητα. Ο δέκτης ή η συσκευή που λαμβάνει τα δεδομένα διαθέτει ένα ενσωματωμένο σύστημα που διευκολύνει την εφαρμογή ενός παγκόσμιου συστήματος εντοπισμού θέσης. Οι ενσωματωμένες συσκευές GPS επιτρέπουν στους ανθρώπους να βρίσκουν εύκολα τις τρέχουσες τοποθεσίες και τους προορισμούς τους. Έτσι, αποκτούν ραγδαία δυναμική και γίνονται τα πιο ευρέως χρησιμοποιούμενα εργαλεία πλοήγησης για τα αυτοκίνητα.

Συστήματα αυτοκινήτων

Τα ενσωματωμένα συστήματα αυτοκινήτων σχεδιάζονται και εγκαθίστανται για την ενίσχυση της ασφάλειας των αυτοκινήτων. Χάρη στα συστήματα ασφαλείας των οχημάτων, το ποσοστό των νεκρών στην κυκλοφορία έχει μειωθεί κατακόρυφα τα τελευταία χρόνια. Οι αυτοκινητοβιομηχανίες κάνουν το κάτι παραπάνω για να ενισχύσουν τα αυτοκίνητα με συστήματα και αισθητήρες προηγμένης τεχνολογίας, κάτι που δεν είναι δυνατό χωρίς ενσωματωμένα συστήματα. Ορισμένα βασικά παραδείγματα ενός συστήματος ενεργητικής ασφάλειας περιλαμβάνουν τον προσαρμοζόμενο έλεγχο ταχύτητας, την προειδοποίηση βλάβης αυτοκινήτου, την αναγνώριση πεζών, τη βοήθεια συγχώνευσης, τους αερόσακους. Αυτά είναι μερικά από τα χαρακτηριστικά που αναμένεται να μετριάσουν τον κίνδυνο ατυχημάτων και να ενισχύσουν τη ζήτηση για ενσωματωμένα συστήματα σε όλο τον κόσμο.

Ιατρικές συσκευές

Οι ιατρικές συσκευές στις εγκαταστάσεις υγειονομικής περίθαλψης ενσωματώνουν ενσωματωμένα συστήματα εδώ και αρκετό καιρό. Μια νέα κατηγορία ιατρικών συσκευών

χρησιμοποιεί ενσωματωμένα συστήματα για τη θεραπεία ασθενών που χρειάζονται συχνή παρακολούθηση και συνεχή προσοχή στο σπίτι. Αυτά τα συστήματα είναι ενσωματωμένα με αισθητήρες για τη συλλογή δεδομένων που σχετίζονται με την υγεία των ασθενών, όπως ο καρδιακός ρυθμός, ο ρυθμός παλμών ή οι ενδείξεις από τα εμφυτεύματα, τα οποία αποστέλλονται σε ένα σύννεφο όπου ο γιατρός μπορεί να εξετάσει τα δεδομένα του ασθενούς στη συσκευή του ασύρματα.

Ελεγκτές φυσικής κατάστασης

Οι συσκευές παρακολούθησης φυσικής κατάστασης είναι φορητές συσκευές που μπορούν να παρακολουθούν την υγεία σας και να παρακολουθούν δραστηριότητες όπως ο ύπνος, το τρέξιμο και το περπάτημα. Αυτές οι συσκευές χρησιμοποιούν ενσωματωμένα συστήματα για να συλλέγουν δεδομένα σχετικά με τον καρδιακό σας ρυθμό, τη θερμοκρασία του σώματος και τον αριθμό των βημάτων σας, τα οποία αποστέλλονται περαιτέρω σε διακομιστές μέσω WAN όπως το LTE (Long-Term Evolution) ή το GPRS (General Packet Radio Service).

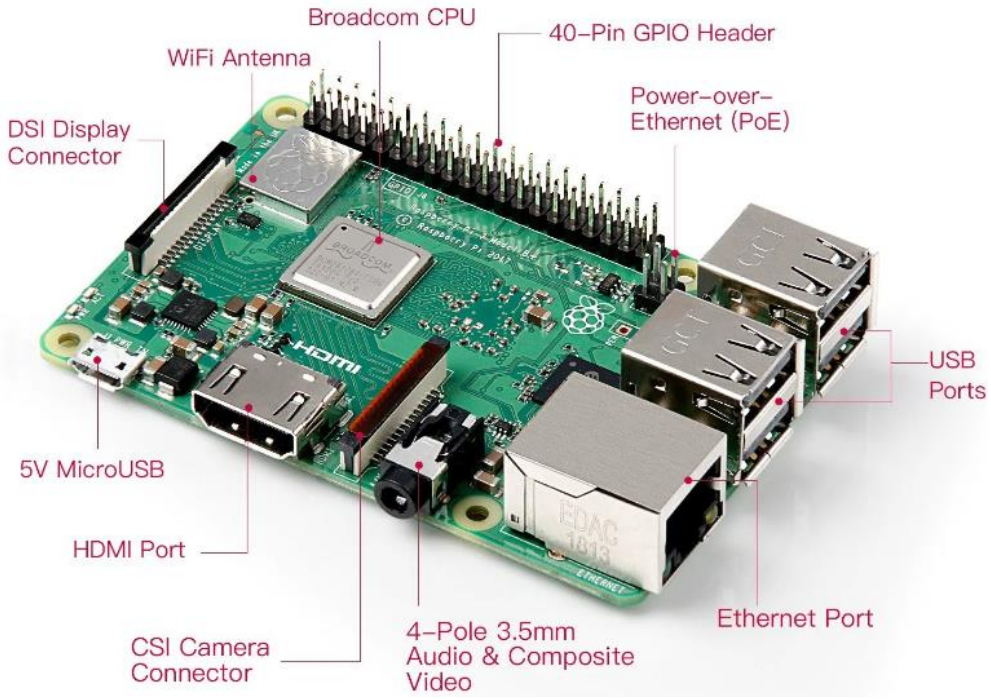
Ρομπότ εργοστασίων

Τα εργοστασιακά ρομπότ έχουν σχεδιαστεί για να εκτελούν εργασίες υψηλής ακρίβειας υπό επικίνδυνες συνθήκες εργασίας. Διαθέτουν ένα ολοκληρωμένο ενσωματωμένο σύστημα για τη σύνδεση διαφόρων υποσυστημάτων. Σε μια τυπική μηχανική εργασία, τα ρομπότ χρησιμοποιούν ενεργοποιητές, αισθητήρες και λογισμικό για να αντιλαμβάνονται το περιβάλλον και να εξάγουν την προβλεπόμενη έξοδο με ασφάλεια. Χωρίς ενσωματωμένο σύστημα, τα ρομπότ θα έπρεπε να βασίζονται σε εξωτερικά συστήματα ελέγχου ή υπολογισμού. Αυτό, με τη σειρά του, μπορεί να αυξήσει τους κινδύνους ασφάλειας λόγω καθυστέρησης ή αποτυχίας στη σύνδεση μεταξύ του εργοστασιακού ρομπότ και του εξωτερικού υπολογιστικού συστήματος.

1.10 Αναπτυξιακές πλακέτες

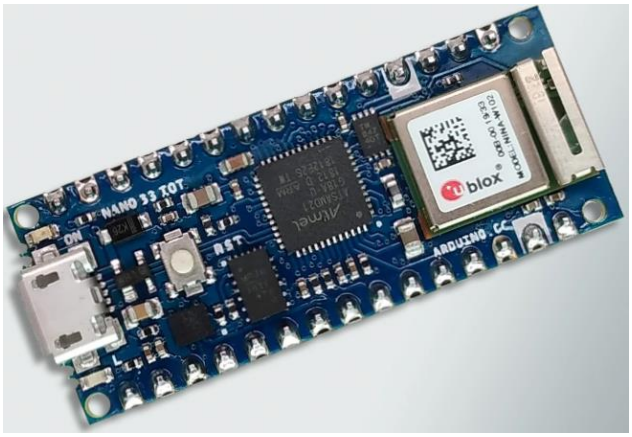
Η πλακέτα ανάπτυξης μικροελεκτών είναι το βασικό συστατικό μιας ενσωματωμένης εφαρμογής ή μιας εφαρμογής IoT, επομένως καθίσταται σημαντικά σημαντική η επιλογή της πλακέτας πριν από το σχεδιασμό της εφαρμογής.

Raspberry Pi



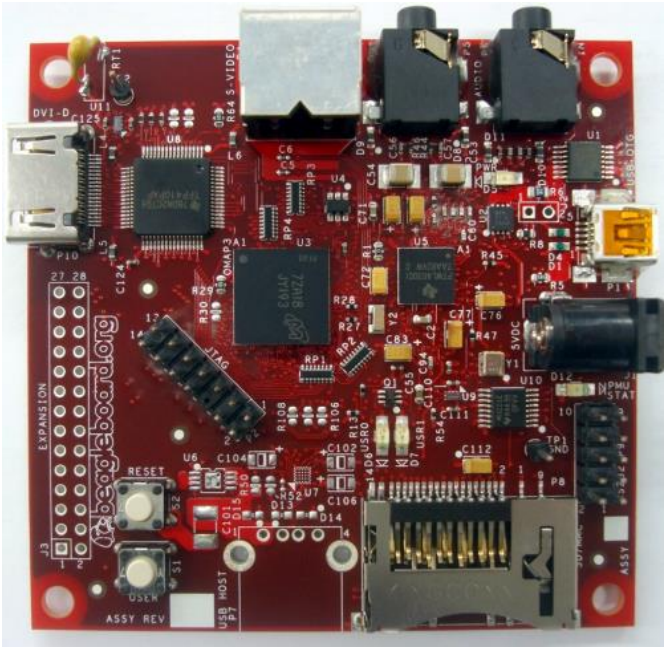
Raspberry Pi είναι η ονομασία μιας σειράς υπολογιστών μιας πλακέτας που κατασκευάζονται από το Ίδρυμα Raspberry Pi, μια βρετανική φιλανθρωπική οργάνωση που έχει ως στόχο να εκπαιδεύσει τους ανθρώπους στην πληροφορική και να δημιουργήσει ευκολότερη πρόσβαση στην εκπαίδευση στην πληροφορική. Το Raspberry Pi κυκλοφόρησε το 2012 και έκτοτε έχουν κυκλοφορήσει αρκετές επαναλήψεις και παραλλαγές. Το αρχικό Pi είχε μονοπύρηνη CPU 700MHz και μόλις 256MB RAM, ενώ το τελευταίο μοντέλο έχει τετραπύρηνη CPU που χρονίζει πάνω από 1,5GHz και 4GB RAM. Η τιμή του Raspberry Pi ήταν πάντα κάτω από 100 δολάρια (συνήθως γύρω στα 35 δολάρια ΗΠΑ), με πιο χαρακτηριστικό το Pi Zero, το οποίο κοστίζει μόλις 5 δολάρια.

Arduino Nano 33 IoT



Το Arduino Nano 33 IoT είναι το ευκολότερο και φθηνότερο σημείο εισόδου για την ενίσχυση των υφιστάμενων συσκευών (και τη δημιουργία νέων) ώστε να γίνουν μέρος του IoT και να σχεδιάσετε εφαρμογές πικο-δικτύου. Είτε θέλετε να δημιουργήσετε ένα δίκτυο αισθητήρων συνδεδεμένο με τον δρομολογητή του γραφείου ή του σπιτιού σας, είτε θέλετε να δημιουργήσετε μια συσκευή Bluetooth® Low Energy που στέλνει δεδομένα σε ένα κινητό τηλέφωνο, το Nano 33 IoT είναι η λύση μιας στάσης για πολλά από τα βασικά σενάρια εφαρμογών IoT. Ο κύριος επεξεργαστής της πλακέτας είναι ένας Arm® Cortex®-M0 32-bit SAMD21 χαμηλής κατανάλωσης ενέργειας. Η συνδεσιμότητα WiFi και Bluetooth® πραγματοποιείται με μια μονάδα της u-blox, το NINA-W10, ένα chipset χαμηλής ισχύος που λειτουργεί στην περιοχή 2,4 GHz. Επιπλέον, η ασφαλής επικοινωνία εξασφαλίζεται μέσω του τσιπ κρυπτογράφησης Microchip® ECC608.

Beagle Board



Το BeagleBoard είναι ένας υπολογιστής μιας πλακέτας που κατασκευάζεται από την Texas Instruments και παρουσιάστηκε το 2008. Σχεδιάστηκε για χομπίστες και ως εκπαιδευτικό εργαλείο για την ανάπτυξη λογισμικού ανοικτού κώδικα. Χρησιμοποιεί μια CPU ARM Cortex-A8 που λειτουργεί σε ταχύτητες έως και 1 GHz και μπορεί να διαμορφωθεί με μνήμη RAM μεταξύ 128 MB και 512 MB. Έχει διαστάσεις 7,5 mm σε κάθε πλευρά και διαθέτει όλες τις λειτουργίες ενός βασικού υπολογιστή. Το BeagleBoard μπορεί να τρέξει έναν αυξανόμενο κατάλογο λειτουργικών συστημάτων, συμπεριλαμβανομένων των Linux, FreeBSD, OpenBSD, RISC OS και SymbianOS, με υποστήριξη για Android σε ενεργή ανάπτυξη. Διαθέτει θύρα USB, ενσωματωμένη GPU που υποστηρίζει επιτάχυνση 3D OpenGL ES και δύο υποδοχές στερεοφωνικού ήχου για είσοδο και έξοδο ήχου. Τα νεότερα μοντέλα περιλαμβάνουν επίσης ενσωματωμένη δικτύωση Wi-Fi.

1.11.1 Εισαγωγή στα λειτουργικά συστήματα

Λειτουργικό σύστημα είναι ένα επίπεδο λογισμικού που διαχειρίζεται τους πόρους του συστήματος και παρέχει στα προγράμματα χρήστη ένα καλύτερο και πιο σαφές μοντέλο υπολογιστή. Το λειτουργικό σύστημα είναι σχεδιασμένο με τέτοιο τρόπο ώστε να μπορεί να διαχειρίζεται τους συνολικούς πόρους και τις λειτουργίες του υπολογιστή. Πρόκειται για ένα πλήρως ολοκληρωμένο σύνολο εξειδικευμένων προγραμμάτων που χειρίζονται όλες τις λειτουργίες του υπολογιστή. Ελέγχει και παρακολουθεί την εκτέλεση όλων των άλλων προγραμμάτων που βρίσκονται στον υπολογιστή, τα οποία περιλαμβάνουν επίσης προγράμματα εφαρμογών και άλλο λογισμικό συστήματος του υπολογιστή. Παραδείγματα λειτουργικού συστήματος είναι τα Windows, το Linux, το Mac OS κ.λπ.

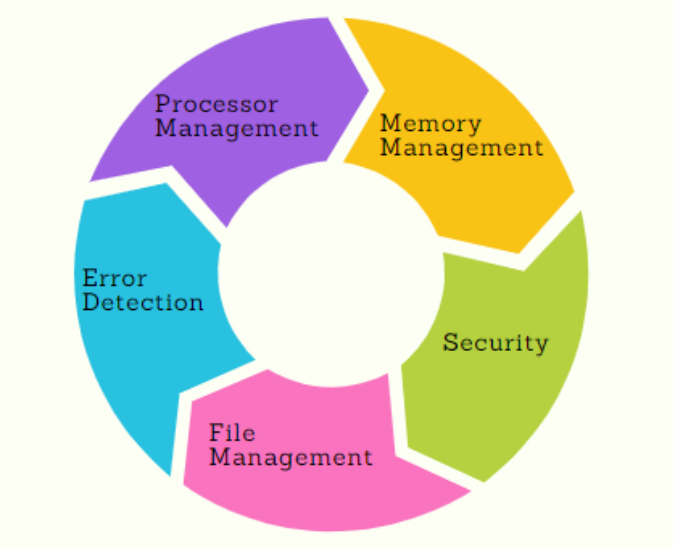
1.11.2 Βασικές έννοιες και λειτουργίες

Κάθε σύστημα υπολογιστή περιλαμβάνει ένα βασικό σύνολο προγραμμάτων που ονομάζεται λειτουργικό σύστημα. Το σημαντικότερο πρόγραμμα του συνόλου ονομάζεται πυρήνας. Φορτώνεται στη μνήμη RAM κατά την εκκίνηση του συστήματος και περιέχει πολλές κρίσιμες διαδικασίες που απαιτούνται για τη λειτουργία του συστήματος. Τα άλλα προγράμματα είναι λιγότερο κρίσιμα βοηθητικά προγράμματα- μπορούν να παρέχουν μια μεγάλη ποικιλία δια δραστικών εμπειριών για τον χρήστη - καθώς και να κάνουν όλες τις εργασίες για τις οποίες ο χρήστης αγόρασε τον υπολογιστή - αλλά η βασική μορφή και οι δυνατότητες του συστήματος καθορίζονται από τον πυρήνα. Ο πυρήνας παρέχει βασικές διευκολύνσεις σε οτιδήποτε άλλο στο σύστημα και καθορίζει πολλά από τα χαρακτηριστικά του ανώτερου λογισμικού. Ως εκ τούτου, χρησιμοποιούμε συχνά τον όρο "λειτουργικό σύστημα" ως συνώνυμο του "πυρήνα". Το λειτουργικό σύστημα πρέπει να εκπληρώνει δύο κύριους στόχους:

-Αλληλεπίδραση με τα στοιχεία υλικού, εξυπηρετώντας όλα τα προγραμματιζόμενα στοιχεία χαμηλού επιπέδου που περιλαμβάνονται στην πλατφόρμα υλικού.

-Παροχή ενός περιβάλλοντος εκτέλεσης στις εφαρμογές που εκτελούνται στο σύστημα υπολογιστή (τα λεγόμενα προγράμματα χρήστη).

Λειτουργίες



Processor Management(Διαχείριση επεξεργαστή): Ένα λειτουργικό σύστημα διαχειρίζεται την εργασία του επεξεργαστή, κατανέμοντας διάφορες εργασίες σε αυτόν και διασφαλίζοντας ότι κάθε διεργασία λαμβάνει αρκετό χρόνο από τον επεξεργαστή για να λειτουργήσει σωστά.

Memory Management (Διαχείριση μνήμης): Ένα λειτουργικό σύστημα διαχειρίζεται την κατανομή και την αποδέσμευση της μνήμης σε διάφορες διεργασίες και διασφαλίζει ότι η άλλη διεργασία δεν καταναλώνει τη μνήμη που έχει κατανεμηθεί σε μια διεργασία.

File Management (Διαχείριση αρχείων): Ένα λειτουργικό σύστημα παρακολουθεί τις πληροφορίες σχετικά με τη δημιουργία, τη διαγραφή, τη μεταφορά, την αντιγραφή και την αποθήκευση αρχείων με οργανωμένο τρόπο. Διατηρεί επίσης την ακεραιότητα των δεδομένων που είναι αποθηκευμένα σε αυτά τα αρχεία, συμπεριλαμβανομένης της δομής καταλόγου αρχείων, προστατεύοντας από μη εξουσιοδοτημένη πρόσβαση.

Security(Ασφάλεια): Το λειτουργικό σύστημα παρέχει διάφορες τεχνικές που εξασφαλίζουν την ακεραιότητα και την εμπιστευτικότητα των δεδομένων του χρήστη.

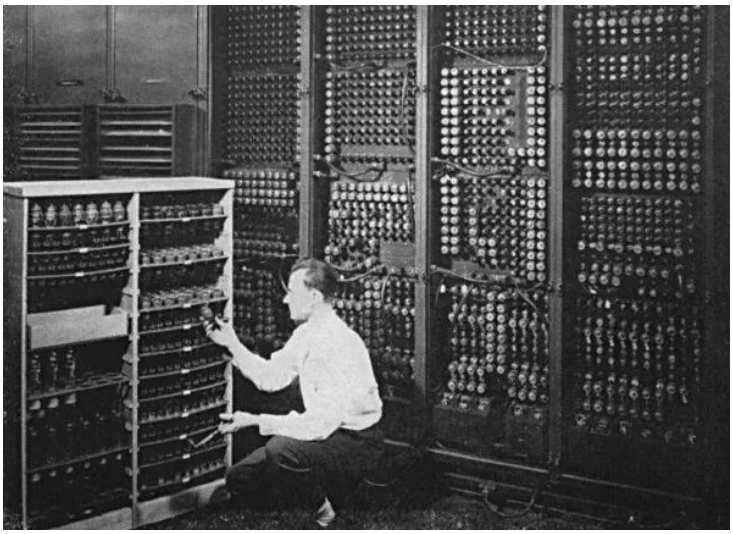
Error Detection(Ανίχνευση σφαλμάτων): Κατά διαστήματα, το λειτουργικό σύστημα ελέγχει το σύστημα για τυχόν εξωτερική απειλή ή δραστηριότητα κακόβουλου λογισμικού. Ελέγχει επίσης

το υλικό για κάθε είδους βλάβη. Αυτή η διαδικασία εμφανίζει διάφορες ειδοποιήσεις στον χρήστη, ώστε να μπορεί να λάβει τα κατάλληλα μέτρα κατά οποιασδήποτε βλάβης που προκαλείται στο σύστημα.

1.11.3 Η εξέλιξη των λειτουργικών συστημάτων

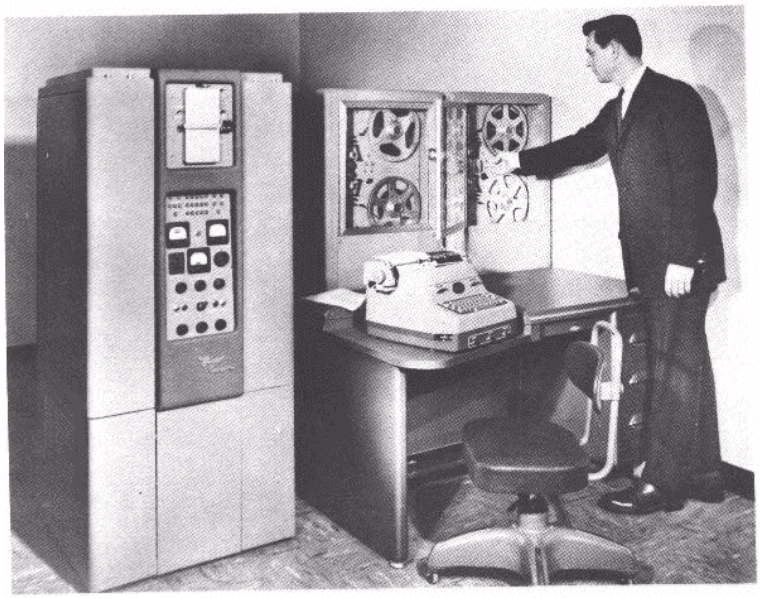
Ιστορικά, τα λειτουργικά συστήματα είναι στενά συνδεδεμένα με την αρχιτεκτονική των υπολογιστών. Είναι καλή ιδέα να μελετήσουμε την ιστορία των λειτουργικών συστημάτων από την αρχιτεκτονική των υπολογιστών στους οποίους τρέχουν. Τα λειτουργικά συστήματα έχουν εξελιχθεί μέσα από μια σειρά διακριτών φάσεων ή γενεών που αντιστοιχούν περίπου στις δεκαετίες. Για να βρούμε τα πρώτα λειτουργικά συστήματα πρέπει να ταξιδέψουμε στη δεκαετία του '50 του 20ου αιώνα. Προηγουμένως, κατά τη δεκαετία του 1940, τα προγράμματα εισάγονταν απευθείας στο υλικό της μηχανής μέσω μιας σειράς μικροδιακοπτών. Στη δεκαετία του 1950 εμφανίστηκαν κάποιες τεχνολογίες που επέτρεπαν μια πιο "απλή" αλληλεπίδραση μεταξύ του χρήστη και του υπολογιστή.

Η πρώτη γενιά (1940 έως αρχές της δεκαετίας του 1950)



Έτσι λοιπόν, Όταν αναπτύχθηκε ο πρώτος ηλεκτρονικός υπολογιστής το 1940, δημιουργήθηκε χωρίς κανένα λειτουργικό σύστημα. Στην αρχή, οι χρήστες έχουν πλήρη πρόσβαση στη μηχανή του υπολογιστή και γράφουν ένα πρόγραμμα για κάθε εργασία σε απόλυτη γλώσσα μηχανής. Ο προγραμματιστής μπορεί να εκτελέσει και να επιλύσει μόνο απλούς μαθηματικούς υπολογισμούς κατά τη διάρκεια της δημιουργίας του υπολογιστή και αυτός ο υπολογισμός δεν απαιτεί λειτουργικό σύστημα.

Η δεύτερη γενιά (1955 - 1965)



Το πρώτο λειτουργικό σύστημα (OS) δημιουργήθηκε στις αρχές της δεκαετίας του 1950 και ήταν γνωστό ως GMOS. Η General Motors ανέπτυξε λειτουργικό σύστημα για τον υπολογιστή IBM. Το λειτουργικό σύστημα δεύτερης γενιάς βασιζόταν σε ένα σύστημα επεξεργασίας παρτίδων μονής ροής, επειδή συλλέγει όλες τις παρόμοιες εργασίες σε ομάδες ή παρτίδες και στη συνέχεια υποβάλλει τις εργασίες στο λειτουργικό σύστημα χρησιμοποιώντας μια κάρτα διάτρησης για να ολοκληρωθούν όλες οι εργασίες σε μια μηχανή. Σε κάθε ολοκλήρωση των εργασιών (είτε κανονικά είτε μη κανονικά), ο έλεγχος μεταφέρεται στο λειτουργικό σύστημα το οποίο καθαρίζεται μετά την ολοκλήρωση μιας εργασίας και στη συνέχεια συνεχίζει να διαβάζει και να ξεκινά την επόμενη εργασία σε μια κάρτα διάτρησης. Μετά από αυτό, τα νέα μηχανήματα ονομάστηκαν mainframes, τα οποία ήταν πολύ μεγάλα και χρησιμοποιούνταν από επαγγελματίες χειριστές.

Η τρίτη γενιά (1965 - 1980)

Στα τέλη της δεκαετίας του 1960, οι σχεδιαστές λειτουργικών συστημάτων ήταν πολύ ικανοί να αναπτύξουν ένα νέο λειτουργικό σύστημα που θα μπορούσε να εκτελεί ταυτόχρονα πολλαπλές εργασίες σε ένα μόνο πρόγραμμα υπολογιστή, το λεγόμενο πολυπρογραμματισμό. Η εισαγωγή του πολυπρογραμματισμού παίζει πολύ σημαντικό ρόλο στην ανάπτυξη λειτουργικών συστημάτων που επιτρέπουν σε μια CPU να είναι απασχολημένη κάθε φορά εκτελώντας ταυτόχρονα διαφορετικές εργασίες σε έναν υπολογιστή. Κατά τη διάρκεια της τρίτης γενιάς, υπήρξε μια νέα εξέλιξη της πρωτοφανούς ανάπτυξης των μικροϋπολογιστών που ξεκίνησε το 1961 με τον DEC PDP-1. Αυτοί οι PDP οδήγησαν στη δημιουργία των προσωπικών υπολογιστών στην τέταρτη γενιά.

Η τέταρτη γενιά (1980 - Σήμερα)

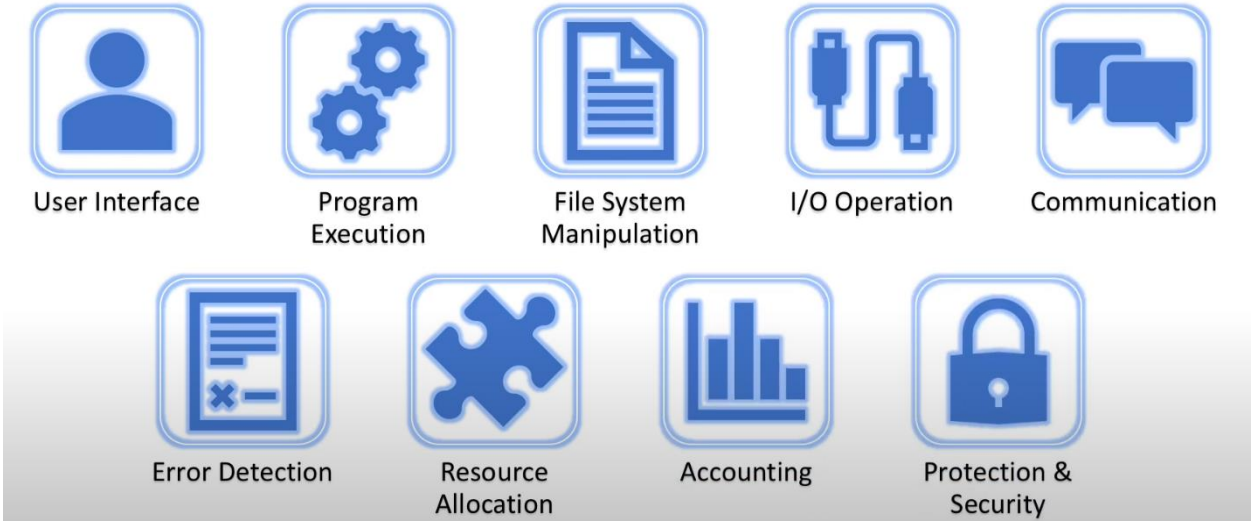
Η τέταρτη γενιά λειτουργικών συστημάτων σχετίζεται με την ανάπτυξη των προσωπικών υπολογιστών. Ωστόσο, ο προσωπικός υπολογιστής μοιάζει πολύ με τους μικροϋπολογιστές που αναπτύχθηκαν στην τρίτη γενιά. Το κόστος ενός προσωπικού υπολογιστή ήταν πολύ υψηλό εκείνη την εποχή- υπήρχαν μικρά κλάσματα του κόστους των μικροϋπολογιστών. Ένας σημαντικός παράγοντας που σχετιζόταν με τη δημιουργία προσωπικών υπολογιστών ήταν η γέννηση της Microsoft και του λειτουργικού συστήματος Windows. Η Microsoft δημιούργησε το πρώτο λειτουργικό σύστημα παραθύρων το 1975. Μετά την παρουσίαση του λειτουργικού συστήματος Microsoft Windows, ο Bill Gates και ο Paul Allen είχαν το όραμα να οδηγήσουν τους

προσωπικούς υπολογιστές στο επόμενο επίπεδο. Ως εκ τούτου, εισήγαγαν το MS-DOS το 1981- ωστόσο, ήταν πολύ δύσκολο για το άτομο να κατανοήσει τις κρυπτογραφημένες εντολές του. Σήμερα, τα Windows έχουν γίνει η πιο δημοφιλής και πιο συχνά χρησιμοποιούμενη τεχνολογία λειτουργικού συστήματος. Και στη συνέχεια, τα Windows κυκλοφόρησαν διάφορα λειτουργικά συστήματα, όπως τα Windows 95, τα Windows 98, τα Windows XP και το πιο πρόσφατο λειτουργικό σύστημα, τα Windows 7. Επί του παρόντος, οι περισσότεροι χρήστες των Windows χρησιμοποιούν το λειτουργικό σύστημα Windows 10. Εκτός από το λειτουργικό σύστημα Windows, η Apple είναι ένα άλλο δημοφιλές λειτουργικό σύστημα που κατασκευάστηκε τη δεκαετία του 1980 και αυτό το λειτουργικό σύστημα αναπτύχθηκε από τον Steve Jobs.

1.11.4 Σκοποί

Οι περισσότεροι άνθρωποι χρησιμοποιούν υπολογιστές επειδή θέλουν να χρησιμοποιούν λογισμικό εφαρμογών (όπως επεξεργασία κειμένου, περιήγηση στο διαδίκτυο), ωστόσο οι εφαρμογές αυτές λειτουργούν μόνο μέσα σε ένα περιβάλλον λειτουργικού συστήματος.Πιο συγκεκριμένα, σκοπός των λειτουργικών συστημάτων είναι να διαχειρίζονται τη μνήμη του υπολογιστή, τις διεργασίες και τη λειτουργία όλου του υλικού και του λογισμικού. Το λειτουργικό σύστημα είναι το πιο σημαντικό λογισμικό σε έναν υπολογιστή, καθώς επιτρέπει στο υλικό του υπολογιστή να επικοινωνεί αποτελεσματικά με όλο το υπόλοιπο λογισμικό του υπολογιστή. Επιπλέον, πρέπει να γνωρίζει τη μονάδα διαχείρισης μνήμης, τη μονάδα διαχείρισης ισχύος, το τμήμα κατάστασης εργασιών και άλλες μονάδες. Με τη χρήση του επεξεργαστή, το λειτουργικό σύστημα χειρίζεται όλα αυτά τα λογικά μέρη και τα αρχικοποιεί.

1.11.5 Υπηρεσίες



User Interface (Διεπαφή χρήστη)

Η διεπαφή χρήστη είναι η υπηρεσία που πρακτικά επιτρέπει στους χρήστες να αλληλοεπιδρούν με ένα λειτουργικό σύστημα. Έτσι, σημαίνει αλληλεπίδραση με ολόκληρο το ίδιο το σύστημα του υπολογιστή. Η διεπαφή χρήστη των λειτουργικών συστημάτων έχει δύο κοινές μορφές: τη διεπαφή γραμμής εντολών (Command Line Interface - CLI) και τη γραφική διεπαφή χρήστη (Graphical User Interface - GUI).

Program Execution (Εκτέλεση προγράμματος)

Το Λειτουργικό Σύστημα είναι αυτό που διαχειρίζεται τον τρόπο εκτέλεσης ενός προγράμματος. Φορτώνει το πρόγραμμα στη μνήμη και στη συνέχεια εκτελείται. Η σειρά με την οποία εκτελούνται εξαρτάται από τους αλγόριθμους χρονοπρογραμματισμού της CPU. Μερικοί από αυτούς είναι οι FCFS, SJF κ.λπ. Όταν το πρόγραμμα βρίσκεται σε εκτέλεση, το Λειτουργικό Σύστημα χειρίζεται επίσης το αδιέξοδο, δηλαδή να μην έρχονται για εκτέλεση δύο διεργασίες ταυτόχρονα. Το Λειτουργικό Σύστημα είναι υπεύθυνο για την ομαλή εκτέλεση τόσο των προγραμμάτων του χρήστη όσο και του συστήματος. Το Λειτουργικό Σύστημα χρησιμοποιεί διάφορους διαθέσιμους πόρους για την αποτελεσματική εκτέλεση όλων των τύπων λειτουργιών.

File System Manipulation (Χειρισμός Συστήματος Αρχείων)

Ένα αρχείο αντιπροσωπεύει μια συλλογή σχετικών πληροφοριών. Οι υπολογιστές μπορούν να αποθηκεύουν αρχεία στο δίσκο (δευτερεύουσα αποθήκευση), για σκοπούς μακροχρόνιας αποθήκευσης. Παραδείγματα μέσων αποθήκευσης περιλαμβάνουν μαγνητική ταινία, μαγνητικό δίσκο και οπτικούς δίσκους όπως CD, DVD. Κάθε ένα από αυτά τα μέσα έχει τις δικές του ιδιότητες, όπως ταχύτητα, χωρητικότητα, ρυθμό μεταφοράς δεδομένων και μεθόδους πρόσβασης στα δεδομένα. Ένα σύστημα αρχείων είναι συνήθως οργανωμένο σε καταλόγους για εύκολη πλοήγηση και χρήση. Αυτοί οι κατάλογοι μπορεί να περιέχουν αρχεία και άλλες κατευθύνσεις.

I/O Operation (Λειτουργία I/O)

Το λειτουργικό σύστημα διαχειρίζεται τις λειτουργίες εισόδου-εξόδου και εγκαθιστά την επικοινωνία μεταξύ του χρήστη και των προγραμμάτων οδήγησης συσκευών. Τα προγράμματα οδήγησης συσκευών είναι λογισμικό που σχετίζεται με υλικό το οποίο διαχειρίζεται το λειτουργικό σύστημα, ώστε να λειτουργεί σωστά ο συγχρονισμός μεταξύ των συσκευών. Παρέχει επίσης πρόσβαση στις συσκευές εισόδου-εξόδου σε ένα πρόγραμμα όταν χρειάζεται.

Communication (Επικοινωνία)

Στην περίπτωση των κατανεμημένων συστημάτων, τα οποία αποτελούν μια συλλογή επεξεργαστών που δεν μοιράζονται μνήμη, περιφερειακές συσκευές ή ρολόι, το λειτουργικό σύστημα διαχειρίζεται τις επικοινωνίες μεταξύ όλων των διεργασιών. Πολλαπλές διεργασίες

επικοινωνούν μεταξύ τους μέσω γραμμών επικοινωνίας στο δίκτυο. Το λειτουργικό σύστημα διαχειρίζεται τις στρατηγικές δρομολόγησης και σύνδεσης, καθώς και τα προβλήματα ανταγωνισμού και ασφάλειας.

Error Detection (Ανίχνευση σφαλμάτων)

Κατά τη διάρκεια του κύκλου ζωής ενός συστήματος υπολογιστή, μπορούν να εμφανιστούν διάφορα σφάλματα. Για παράδειγμα, μπορεί να έχουμε σφάλματα CPU, κακές κατανομές ή προσπελάσεις μνήμης, βλάβη εξαρτήματος που προκαλεί σφάλμα υλικού ή ακόμη και σφάλμα που προκαλείται από μια συσκευή εισόδου/εξόδου. Πρώτον, η υπηρεσία ανίχνευσης σφαλμάτων των λειτουργικών συστημάτων πρέπει να αποφεύγει την πλήρη κατάρρευση ενός συστήματος υπολογιστή όταν συμβαίνει ένα σφάλμα. Έτσι, η υπηρεσία αυτή πρέπει να πιάνει τα σφάλματα και να τα διαχειρίζεται, διατηρώντας ολόκληρο το σύστημα όσο το δυνατόν πιο λειτουργικό. Επιπλέον, η υπηρεσία ανίχνευσης σφαλμάτων πρέπει να είναι σε θέση να ενημερώνει τον χρήστη για τα σφάλματα, παρουσιάζοντας τους κωδικούς τους, τις περιγραφές τους και, αν είναι γνωστό, τους τρόπους διόρθωσής τους.

Resource Allocation (Κατανομή πόρων)

Η κατανομή πόρων σημαίνει την αφιέρωση υπολογιστικών πόρων σε διεργασίες και χρήστες. Έχουμε πολλούς τύπους πόρων, όπως η CPU (στην πραγματικότητα, ο χρόνος της CPU), η μνήμη, η δικτύωση και οι συσκευές εισόδου/εξόδου. Έτσι, καθώς το λειτουργικό σύστημα ελέγχει αυτούς τους πόρους, φυσικά αποφασίζει ποιες διεργασίες τους χρησιμοποιούν. Κατά τη διάρκεια του κύκλου ζωής της, μια διεργασία τυπικά απαιτεί πολλαπλούς υπολογιστικούς πόρους. Υπάρχουν δύο κύριες προκλήσεις στη διαχείριση πόρων: (i) η αποφυγή του να μην πάρει ποτέ μια διεργασία που απαιτεί έναν συγκεκριμένο πόρο- (ii) η αποφυγή του να μην τον αποδεσμεύσει ποτέ μια διεργασία που έχει έναν πόρο.

Accounting (Λογιστική)

Η λογιστική συνίσταται στην παρακολούθηση της συμπεριφοράς τόσο των χρηστών όσο και των διεργασιών στο υπολογιστικό σύστημα. Για παράδειγμα, το λειτουργικό σύστημα αναλύει ποιοι χρήστες απαιτούν την εκτέλεση ποιων διεργασιών. Έτσι, μπορεί επίσης να διερευνήσει πόσοι υπολογιστικοί πόροι ζητούνται από τις διεργασίες που εκτελούνται στον υπολογιστή. Τέλος, το λειτουργικό σύστημα μπορεί να συσχετίσει τις ληφθείσες πληροφορίες και να δημιουργήσει στατιστικά στοιχεία σχετικά με τους χρήστες του υπολογιστικού συστήματος και τις διεργασίες που εκτελούνται σε αυτό.

Protection & Security (Προστασία & Ασφάλεια)

Λαμβάνοντας υπόψη ένα σύστημα υπολογιστών με πολλούς χρήστες και ταυτόχρονη εκτέλεση πολλών διεργασιών, οι διάφορες διεργασίες πρέπει να προστατεύονται από τις δραστηριότητες των άλλων. Η προστασία αναφέρεται σε έναν μηχανισμό ή έναν τρόπο ελέγχου της πρόσβασης των προγραμμάτων, των διεργασιών ή των χρηστών στους πόρους που ορίζονται από ένα σύστημα υπολογιστή.

1.11.6 Είδη

- Λ.Σ Πολυμέσων

Βασική έννοια η υποστήριξη βίντεο και ήχου. Ιδιαιτερότητες στον χρονοπρογραμματισμό / χρονομερισμό για την ευαισθησία στον χρονισμό παρουσίασης.

- Λ.Σ Πραγματικού χρόνου

Το χρονικό διάστημα που χρειάζεται ένα λειτουργικό σύστημα πραγματικού χρόνου για να επεξεργαστεί και να ανταποκριθεί στις εισόδους είναι πολύ μικρό.

- Μοντέρνα συστήματα

Αυτά τα συστήματα εκμεταλλεύονται κυρίως τις δυνατότητες ενός δικτύου υπολογιστών.

- Λ.Σ Δικτύων

Σε αυτά τα συστήματα το Λ.Σ. δίνει την δυνατότητα επικοινωνίας με άλλες μηχανές συνδεδεμένες με το ίδιο δίκτυο. Για παράδειγμα, διεργασίες σε έναν υπολογιστή μπορούν να ζητήσουν αρχεία να μεταφερθούν από έναν άλλον υπολογιστή. Ακόμα μπορούν να κάνουν “remote login” σε άλλους υπολογιστές και να χρησιμοποιήσουν τους πόρους τους.

- Κατανεμημένα Λ.Σ.

Τα κατανεμημένα Λ.Σ. παρέχουν επίσης τις παραπάνω δυνατότητες. Αλλά με τρόπο διαφανή, δηλαδή ο χρήστης δεν χρειάζεται να γνωρίζει ποιο αρχείο έχει αποθηκευτεί σε ποιον υπολογιστή κ.λπ. ή γενικά να γνωρίζει τίποτα περί κατανομής. Έτσι φαίνεται ότι το Λ.Σ. είναι ένα κεντρικό και όχι κατανεμημένο Λ.Σ. Ο τρόπος πρόσβασης σε όλους τους υλικούς και λογισμικούς πόρους είναι ανεξάρτητος και κατανοητός.

1.11.7 Πυρήνας

Ο όρος πυρήνας (kernel) αναφέρεται στο βασικό, θεμέλιο τμήμα ενός λειτουργικού συστήματος το οποίο αλληλοεπιδρά στενά με το υλικό του υπολογιστή. Οι πυρήνες χωρίζονται σε διαφορετικούς τύπους με βάση το σχεδιασμό / υλοποίηση και τον τρόπο εκτέλεσης κάθε εργασίας του λειτουργικού συστήματος. Όλος ο κωδικός συστήματος εκτελείται στον ίδιο χώρο διευθύνσεων (για λόγους βελτίωσης της απόδοσης) από τους μονολιθικούς πυρήνες. Όμως, οι περισσότερες υπηρεσίες εκτελούνται στο χώρο του χρήστη από τα μικροπύρηνα (η συντήρηση και η αρθρωτότητα μπορούν να αυξηθούν με αυτήν την προσέγγιση). Υπάρχουν πολλές άλλες προσεγγίσεις μεταξύ αυτών των δύο άκρων.

Ο πυρήνας του Λ.Σ. αναφέρεται στο κύριο τμήμα του Λ.Σ. το οποίο υλοποιεί τις δύο βασικές οντότητες του Λ.Σ. Αυτός ο κώδικας είναι προστατευμένος, με την έννοια ότι δεν ανήκει σε κανένα χρήστη. Διαφορετικά, ο κάθε χρήστης θα μπορούσε να αλλάξει αυτόν τον κώδικα και έτσι να μονοπωλήσει τους πόρους του συστήματος ,CPU,RAM, δίσκοι, τερματικά, κ.λπ.

1.11.8 Διεργασία

Μια διεργασία είναι ένα πρόγραμμα σε εκτέλεση, το οποίο αποτελεί τη βάση όλων των υπολογισμών. Η διεργασία δεν είναι το ίδιο με τον κώδικα του προγράμματος αλλά πολύ περισσότερο από αυτόν. Μια διεργασία είναι μια "ενεργή" οντότητα σε αντίθεση με το

πρόγραμμα που θεωρείται "παθητική" οντότητα. Τα χαρακτηριστικά που κατέχει η διεργασία περιλαμβάνουν την κατάσταση του υλικού, τη μνήμη, τη CPU κ.λπ.

Η μνήμη της διεργασίας χωρίζεται σε τέσσερα τμήματα για αποτελεσματική εργασία:

-Stack: Η στοίβα αποθηκεύει προσωρινά δεδομένα όπως παραμέτρους συναρτήσεων, διευθύνσεις επιστροφών και τοπικές μεταβλητές.

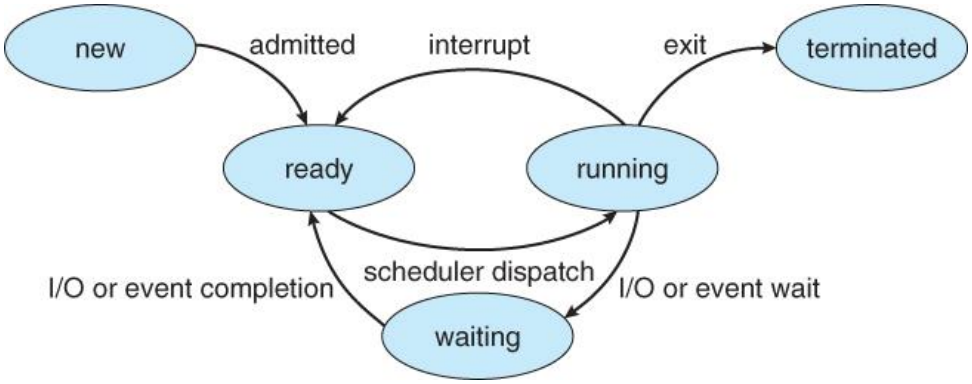
-Heap: Διαθέτει μνήμη, η οποία μπορεί να υποστεί επεξεργασία κατά τη διάρκεια της εκτέλεσης της.

-Data: Περιέχει τη μεταβλητή.

-Κείμενο: Το τμήμα κειμένου περιλαμβάνει την τρέχουσα δραστηριότητα, η οποία αντιπροσωπεύεται από την τιμή του μετρητή προγράμματος.

Καταστάσεις μιας διαδικασίας

Κατά τη διάρκεια της εκτέλεσης μιας διεργασίας, αυτή περνάει από διάφορες καταστάσεις και αλλάζει την κατάστασή της όσες φορές απαιτείται. Έτσι, οι διεργασίες στο λειτουργικό σύστημα μπορεί να βρίσκονται σε οποιαδήποτε από τις ακόλουθες καταστάσεις:



New: Η διεργασία δημιουργείται

Running: Η διεργασία εκτελείται σε κάποιον επεξεργαστή

Waiting: Η διεργασία αναμένει (απενεργοποιημένη) κάποιο συμβάν

Ready: Η διεργασία αναμένει να της δοθεί (από το ΛΣ) σε κάποιον επεξεργαστή για την συνέχιση της εκτέλεσης της

Terminated: Η διεργασία έχει ολοκληρώσει την εκτέλεση της

Μετά τη δημιουργία της, η διεργασία περιμένει μέχρι το λειτουργικό σύστημα να την αναθέσει στον επεξεργαστή. Έτσι, ο επεξεργαστής βρίσκεται σε κατάσταση ετοιμότητας. Η διεργασία αλλάζει την κατάστασή της σε κατάσταση ετοιμότητας μόλις δημιουργηθεί. Ή μερικές φορές η διεργασία διακόπτεται. Και η CPU κατανέμεται σε άλλες διεργασίες. Στη συνέχεια, μετά την εξυπηρέτηση της διακοπής η διεργασία παραμένει και πάλι σε κατάσταση ετοιμότητας.

1.11.9 Χρονοδρομολόγηση

Η ανάθεση της εκτέλεσης των διεργασιών στον επεξεργαστή με σκοπό την ικανοποίηση των παραμέτρων απόδοσης. Την χρονοδρομολόγηση υλοποιεί ειδικό μέρος του ΛΣ που αποκαλείται χρονοδρομολογητής (scheduler).

1.11.9.1 Είδη Χρονοδρομολόγησης

Μακροπρόθεσμη

Αποφασίζει αν μια νέα διεργασία θα προστεθεί στις ενεργές διεργασίες του συστήματος. Διαμορφώνει τον βαθμό πολυπρογραμματισμού.

Μεσοπρόθεσμη

Αποφασίζει αν μια διεργασία θα μεταφερθεί στην κεντρική μνήμη προκειμένου να εκτελεστεί. Διαμορφώνει τον βαθμό πολυπρογραμματισμού και υποστηρίζεται από πολιτικές διαχείρισης μνήμης.

Βραχυπρόθεσμη

Αποφασίζει ποια από τις διεργασίες που βρίσκονται στην μνήμη, θα εκτελεστεί στη συνέχεια. Εφαρμόζεται μετά από το μπλοκάρισμα της τρέχουσας διεργασίας ή όταν η τρέχουσα προ εκχωρεί την ΚΜΕ σε άλλη διεργασία.

1.11.9.2 Αλγόριθμοι Χρονοδρομολόγησης

Πρώτη εισερχόμενη – πρώτη εξυπηρετούμενη (FCFS)

- Οι διεργασίες εξυπηρετούνται με τη σειρά που εισέρχονται.
- Η τρέχουσα κρατά την ΚΜΕ μέχρι είτε να εκτελέσει Ε/Ε είτε να τελειώσει.
- Ο FCFS είναι μη προ εκχωρητικός αλγόριθμος.
- Δίκαιος αλγόριθμος όσον αφορά το κριτήριο του χρόνο άφιξης.
- Άσχημος χρόνος απόκρισης (π.χ. στην περίπτωση μεγάλης διεργασίας που ακολουθείται από μικρές).
- Ευνοεί τις διεργασίες που απαιτούν περισσότερο χρόνο ΚΜΕ παρά χρόνο I/O.

Εκ περιτροπής (Round Robin – RR)

- Θεωρείται η προ εκχωρητική έκδοση του FCFS.
- Ο χρονοδρομολογητής διατηρεί μια ουρά FIFO. Μια διεργασία που εισέρχεται στο σύστημα τοποθετείται στο τέλος της ουράς ενώ η επόμενη προς εκτέλεση επιλέγεται από την αρχή της ουράς. Κάθε διεργασία θα εκτελεστεί για ένα προκαθορισμένο

χρονικό διάστημα (κβάντα χρόνου) και στη συνέχεια θα εκτελεστεί η επόμενη διεργασία της ουράς.

- Εξαιρετικά δίκαιος αλγόριθμος (ίδια κβάντα χρόνου σε όλες τις διεργασίες + FCFS).
- Χρησιμοποιείται στα συστήματα διαμοιρασμού χρόνου.

Η συντομότερη διεργασία πρώτη(shortest job first – SJF)

- Η διεργασία με το μικρότερο χρόνο εκτέλεσης δρομολογείται πρώτη.
- Είναι μη προ εκχωρητικός αλγόριθμος.
- Βελτιστοποιεί το μέσο χρόνο διεκπεραίωσης.
- Δυσκολία υλοποίησης λόγω της δυσκολίας εκτίμησης του χρόνου εκτέλεσης κάθε διεργασία.

1.12.1 Τρόποι εκτέλεσης

Ένα σφάλμα σε ένα πρόγραμμα μπορεί να επηρεάσει αρνητικά πολλές διεργασίες, μπορεί να τροποποιήσει δεδομένα ενός άλλου προγράμματος ή μπορεί επίσης να επηρεάσει το λειτουργικό σύστημα. Για παράδειγμα, αν μια διεργασία κολλήσει σε ατέρμονο βρόχο, τότε αυτός ο ατέρμονος βρόχος μπορεί να επηρεάσει τη σωστή λειτουργία άλλων διεργασιών. Για να διασφαλιστεί λοιπόν η σωστή εκτέλεση του λειτουργικού συστήματος, υπάρχουν δύο τρόποι λειτουργίας:

Λειτουργία χρήστη

Όταν το σύστημα υπολογιστή εκτελεί εφαρμογές χρήστη, όπως η δημιουργία ενός εγγράφου κειμένου ή η χρήση οποιουδήποτε προγράμματος εφαρμογής, τότε το σύστημα βρίσκεται σε κατάσταση λειτουργίας χρήστη. Όταν η εφαρμογή χρήστη ζητάει μια υπηρεσία από το λειτουργικό σύστημα ή εμφανίζεται μια διακοπή ή μια κλήση συστήματος, τότε θα υπάρξει μετάβαση από τη λειτουργία χρήστη στη λειτουργία πυρήνα για την εκπλήρωση των αιτημάτων.

Λειτουργία πυρήνα

Ενώ το σύστημα εκτελείται, ορισμένες διεργασίες λειτουργούν σε λειτουργία πυρήνα, επειδή οι διεργασίες χρειάζονται πρόσβαση σε κλήσεις του λειτουργικού συστήματος. Αυτό παρέχει προστασία ελέγχοντας ποιες διεργασίες μπορούν να έχουν πρόσβαση σε λειτουργίες λειτουργίας πυρήνα. Όπως φαίνεται στο παραπάνω διάγραμμα, το σύστημα επιτρέπει σε ορισμένες διεργασίες λειτουργίας χρήστη να εκτελούν κλήσεις συστήματος, επιτρέποντας στη διεργασία να εκτελείται προσωρινά σε λειτουργία πυρήνα. Ενώ βρίσκεται σε λειτουργία πυρήνα, η διεργασία επιτρέπεται

να έχει άμεση πρόσβαση σε όλο το υλικό και τη μνήμη του συστήματος (ονομάζεται επίσης προνομιακή λειτουργία). Εάν μια διεργασία χρήστη προσπαθήσει να εκτελέσει προνομιούχες εντολές σε κατάσταση λειτουργίας χρήστη, τότε θα αντιμετωπίσει την εντολή ως παράνομη και θα υποβάλει παγίδες στο λειτουργικό σύστημα. Ορισμένες από τις προνομιούχες εντολές είναι οι εξής:

-Χειρισμός διακοπών συστήματος

-Για να μεταβείτε από τη λειτουργία χρήστη στη λειτουργία πυρήνα.

-Διαχείριση συσκευών εισόδου/εξόδου.

1.12.2 Δομή

Τα σύγχρονα Λ.Σ. είναι δομημένα σε ιεραρχικά τοποθετημένα επίπεδα (layers). Κάθε επίπεδο εκτελεί μια συγκεκριμένη εργασία και συνεργάζεται με τα δύο γειτονικά του. Στα κατώτερα επίπεδα γίνεται η διαχείριση της μνήμης και της επικοινωνίας με τις περιφερειακές συσκευές του υπολογιστή, ενώ στα ανώτερα γίνεται η διαχείριση των προγραμμάτων που εκτελούν οι χρήστες. Σε ένα Λ.Σ. υπάρχουν τα ακόλουθα επίπεδα :

α) Τον Πυρήνα (Kernel) : Είναι το κυριότερο τμήμα ενός ΛΣ. Το τμήμα αυτό φορτώνεται πρώτο στην κύρια μνήμη και εκτελείται συνεχώς σε όλη τη διάρκεια λειτουργίας του υπολογιστή. Τα προγράμματα εφαρμογών επικοινωνούν με αυτό μέσα από ένα καθορισμένο σύνολο κλήσεων. Ο πυρήνας είναι ο κύριος υπεύθυνος για τη συνεργασία του λογισμικού με το υλικό του υπολογιστή,

β) Τη διεπαφή χρήστη (user interface) : αναλαμβάνει να δέχεται και να δίνει στο σύστημα του υπολογιστή τα αιτήματα (εντολές) του χρήστη και επίσης να μεταφέρει στο χρήστη μηνύματα από το σύστημα. Το τμήμα αυτό δημιουργεί το περιβάλλον επικοινωνίας χρήστη – υπολογιστή και μπορεί να υλοποιηθεί με περιβάλλον γραμμής εντολών ή με γραφικό περιβάλλον ή και με τους δύο τρόπους.

γ) Το Σύστημα Αρχείων (File Systems) : διαχειρίζεται τα αρχεία (ονοματολογία, καταχώριση, ανάκτηση κ.λπ.) και φροντίζει επίσης για τη διάθεσή τους στους χρήστες.

δ) Τα βοηθητικά προγράμματα (Utilities) : βοηθούν στη διαχείριση του συστήματος.

Κεφάλαιο 2: Αρχιτεκτονική επεξεργαστών

2.1 Εισαγωγή

Οι σύγχρονοι μικροεπεξεργαστές είναι από τα πιο πολύπλοκα συστήματα που δημιούργησε ποτέ ο άνθρωπος. Ένα μόνο πυρίτιο τσιπ, στο μέγεθος περίπου ενός νυχιού, μπορεί να περιέχει έναν πλήρη επεξεργαστή υψηλής απόδοσης, μεγάλη κρυφή μνήμη cache μνήμες και τη λογική που απαιτείται για τη διασύνδεσή του με εξωτερικές συσκευές. Από άποψη επιδόσεων, οι επεξεργαστές που υλοποιούνται σήμερα σε ένα μόνο τσιπ επισκιάζουν τους υπερυπολογιστές

μεγέθους δωματίου που κοστίζουν πάνω από 10 εκατομμύρια δολάρια μόλις πριν από 20 χρόνια. Ακόμη και οι ενσωματωμένοι επεξεργαστές που βρίσκονται σε καθημερινές συσκευές όπως κινητά τηλέφωνα, προσωπικές ψηφιακούς βοηθούς και τα φορητά συστήματα παιχνιδιών είναι πολύ πιο ισχυροί από τους πρώτους προγραμματιστές υπολογιστών οραματίστηκαν ποτέ.

Σε αυτό το κεφάλαιο, θα μελετήσουμε την αρχιτεκτονική καθώς και τα βασικά στοιχεία των επεξεργαστών

Για αρχή ας πούμε λίγα λόγια για τον επεξεργαστή. Η κεντρική μονάδα επεξεργασίας ονομάζεται επίσης επεξεργαστής, κεντρικός επεξεργαστής ή μικροεπεξεργαστής. Εκτελεί όλες τις σημαντικές λειτουργίες ενός υπολογιστή. Λαμβάνει εντολές τόσο από το υλικό όσο και από το ενεργό λογισμικό και παράγει αντίστοιχα αποτελέσματα. Αποθηκεύει όλα τα σημαντικά προγράμματα, όπως τα λειτουργικά συστήματα και το λογισμικό εφαρμογών. Η CPU βοηθά επίσης τις συσκευές εισόδου και εξόδου να επικοινωνούν μεταξύ τους. Λόγω αυτών των χαρακτηριστικών της ΚΜΕ, αναφέρεται συχνά ως ο εγκέφαλος του υπολογιστή.

Η ΚΜΕ εγκαθίσταται ή εισάγεται σε μια υποδοχή ΚΜΕ που βρίσκεται στη μητρική πλακέτα. Επιπλέον, είναι εφοδιασμένη με μια ψήκτρα για την απορρόφηση και τη διάχυση της θερμότητας ώστε να διατηρείται η ΚΜΕ δροσερή και να λειτουργεί ομαλά. Σε γενικές γραμμές, η ΚΜΕ αποτελείται από τρία στοιχεία:

- ALU (Αριθμητική λογική μονάδα)
- Μονάδα ελέγχου
- Μονάδα μνήμης ή αποθήκευσης

ALU: Είναι η αριθμητική λογική μονάδα, η οποία εκτελεί αριθμητικές και λογικές λειτουργίες. Οι αριθμητικές λειτουργίες περιλαμβάνουν την πρόσθεση, την αφαίρεση, τον πολλαπλασιασμό, τη διαίρεση και τις συγκρίσεις. Οι λογικές λειτουργίες περιλαμβάνουν κυρίως την επιλογή, τη σύγκριση και τη συγχώνευση των δεδομένων. Μια CPU μπορεί να περιέχει περισσότερες από μία ALU. Επιπλέον, οι ALU μπορούν να χρησιμοποιηθούν για τη διατήρηση χρονομετρητών που βοηθούν στη λειτουργία του υπολογιστή.

Μονάδα ελέγχου: Είναι το κύκλωμα της μονάδας ελέγχου, το οποίο χρησιμοποιεί ηλεκτρικά σήματα για να καθοδηγήσει το σύστημα του υπολογιστή για την εκτέλεση ήδη αποθηκευμένων εντολών. Λαμβάνει εντολές από τη μνήμη και στη συνέχεια αποκωδικοποιεί και εκτελεί αυτές τις εντολές. Έτσι, ελέγχει και συντονίζει τη λειτουργία όλων των τμημάτων του υπολογιστή. Το κύριο καθήκον της μονάδας ελέγχου είναι να διατηρεί και να ρυθμίζει τη ροή των πληροφοριών στον επεξεργαστή. Δεν συμμετέχει στην επεξεργασία και την αποθήκευση δεδομένων.

Μονάδα μνήμης ή αποθήκευσης/ καταχωρητές: Ονομάζεται μνήμη τυχαίας προσπέλασης (RAM). Αποθηκεύει προσωρινά δεδομένα, προγράμματα και ενδιάμεσα και τελικά

αποτελέσματα της επεξεργασίας. Έτσι, λειτουργεί ως χώρος προσωρινής αποθήκευσης που κρατά προσωρινά τα δεδομένα, τα οποία χρησιμοποιούνται για τη λειτουργία του υπολογιστή.

Οι περισσότερες σύγχρονες CPU υλοποιούνται σε μικροεπεξεργαστές ολοκληρωμένου κυκλώματος (IC), με μία ή περισσότερες CPU σε ένα μόνο chip IC. Τα τσιπ μικροεπεξεργαστών με πολλαπλές ΚΜΕ είναι επεξεργαστές πολλαπλών πυρήνων. Οι μεμονωμένες φυσικές ΚΜΕ, πυρήνες επεξεργαστή, μπορούν επίσης να είναι πολυνηματικές για τη δημιουργία πρόσθετων εικονικών ή λογικών ΚΜΕ.

2.2 Ο επεξεργαστής ARM

Ο επεξεργαστής Arm είναι ένας από μια οικογένεια κεντρικών μονάδων επεξεργασίας (CPU) που βασίζονται στην αρχιτεκτονική RISC (Reduced Instruction Set Computer) για επεξεργαστές υπολογιστών. Η Arm Limited, η εταιρεία που βρίσκεται πίσω από τον επεξεργαστή Arm, σχεδιάζει τα βασικά στοιχεία της CPU και παραχωρεί την πνευματική ιδιοκτησία σε συνεργαζόμενους οργανισμούς, οι οποίοι στη συνέχεια κατασκευάζουν τσιπ με βάση την Arm σύμφωνα με τις δικές τους απαιτήσεις. Η Arm Limited δεν κατασκευάζει ούτε πωλάει απευθείας τσιπ. Η Acorn Computers ανέπτυξε για πρώτη φορά τον επεξεργαστή Arm τη δεκαετία του 1980. Μέχρι πρόσφατα, το όνομα Arm αντιμετωπιζόταν ως ακρωνύμιο, ARM, το οποίο αρχικά σήμαινε Acorn RISC Machine και στη συνέχεια Advanced RISC Machine. Το ακρωνύμιο εξακολουθεί να χρησιμοποιείται ευρέως, αν και η Arm Limited χρησιμοποιεί μόνο το Arm όταν περιγράφει την τεχνολογία των επεξεργαστών της.

Οι επεξεργαστές Arm μπορούν να εκτελέσουν πολύ περισσότερα εκατομμύρια εντολές ανά δευτερόλεπτο από τους επεξεργαστές Intel. Με την αφαίρεση των περιττών εντολών και τη βελτιστοποίηση των διαδρομών, ένας επεξεργαστής Arm μπορεί να παρέχει εξαιρετικές επιδόσεις, ενώ χρησιμοποιεί πολύ λιγότερη ενέργεια από έναν επεξεργαστή που βασίζεται σε CISC. Η μείωση της ενέργειας σημαίνει επίσης ότι οι επεξεργαστές Arm παράγουν λιγότερη θερμότητα. Αυτό δεν σημαίνει ότι οι επεξεργαστές Arm είναι εγγενώς καλύτεροι από τους επεξεργαστές Intel, παρά μόνο ότι είναι καταλληλότεροι για συγκεκριμένες περιπτώσεις χρήσης.

2.2.1 Η ανάπτυξη της αρχιτεκτονικής

Η ARM (Advanced RISC machine) είναι ο πρώτος επεξεργαστής υπολογιστή μειωμένου συνόλου εντολών (RISC) για εμπορική χρήση, ο οποίος αναπτύσσεται επί του παρόντος από την ARM Holdings. Η ιστορία του επεξεργαστή ARM χρονολογείται από το 1983 στην Αγγλία, όταν η Acorn Computers Ltd ξεκίνησε επίσημα το έργο Acorn RISC Management, αφού εμπνεύστηκε να σχεδιάσει τον δικό της επεξεργαστή από το Berkeley RISC, ένα από τα έργα υψηλής απήχησης στο πλαίσιο του έργου VLSI της ARPA (Advanced Research Projects Agency, που τώρα έχει μετατραπεί σε DARPA), το οποίο ασχολείτο με το σχεδιασμό μικροεπεξεργαστών βασισμένων σε RISC, με επικεφαλής τον David Patterson, ο οποίος επινόησε τον όρο "RISC". Στην αρχή ήταν γνωστός ως Acorn RISC machine. Με την VLSI Technology Inc. ως συνεργάτη της σε πυρίτιο, η

ARM παρουσίασε τον ARM1, το πρώτο πυρίτιο ARM στις 26 Απριλίου 1985, το οποίο χρησιμοποιήθηκε ως δεύτερος επεξεργαστής στο BBC Micro για την ανάπτυξη του λογισμικού προσομοίωσης για την ολοκλήρωση των εργασιών στα τσιπ υποστήριξης (VIDC, IOC και MEMC) και για την αύξηση της ταχύτητας λειτουργίας του λογισμικού CAD που χρησιμοποιήθηκε για την ανάπτυξη του ARM2. Η Apple, ενώ ανέπτυσσε μια εντελώς νέα υπολογιστική πλατφόρμα για το Newton, έναν προσωπικό ψηφιακό βοηθό, διαπίστωσε ότι μόνο η μηχανή RISC της Acorn πλησίαζε τις απαιτήσεις που απαιτούνταν για την υλοποίηση, αλλά δεδομένου ότι η ARM δεν είχε ενσωματωμένη μονάδα διαχείρισης μνήμης, η Apple συνεργάστηκε με την Acorn για την ανάπτυξη της ARM.

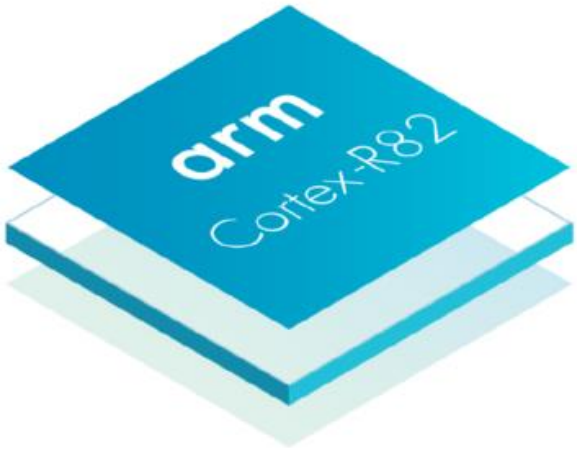
Το αποτέλεσμα αυτής της συνεργασίας ήταν ότι τόσο η Acorn Group όσο και η Apple Computer, Inc. με 43% μερίδιο η καθεμία, καθώς και η VLSI Technology, Inc. ως επενδυτής, ίδρυσαν το 1990 μια ξεχωριστή εταιρεία, την ARM Ltd. Επίσης, το τμήμα προηγμένης έρευνας και ανάπτυξης της Acorn απασχολήθηκε εδώ. Μετά από αυτό το διάστημα, η ARM έγινε το ακρωνύμιο για την προηγμένη μηχανή RISC.

2.2.2 Δημοφιλής επεξεργαστές ARM

Cortex-M23

Ο επεξεργαστής Cortex-M23 είναι ένας πολύ συμπαγής επεξεργαστής με αγωγούς δύο σταδίων που υποστηρίζει το βασικό σύνολο εντολών Arm Version8-M. Ο Cortex-M23 με Trust Zone είναι ο ιδανικός επεξεργαστής για τις πιο περιορισμένες εφαρμογές IoT και ενσωματωμένες εφαρμογές όπου η ασφάλεια αποτελεί βασική απαίτηση.

Cortex-R82



Επεξεργαστής Cortex-R υψηλότερων επιδόσεων με μονάδα διαχείρισης μνήμης (MMU), που επιτρέπει τον πραγματικό χρόνο και πλούσια λειτουργικά συστήματα (OS), όπως το Linux, στον ίδιο πυρήνα ή συστάδα. Ως ο πρώτος επεξεργαστής Arm 64-bit Cortex-R, ο Cortex-R82 μπορεί να απευθύνεται σε μνήμη δυναμικής τυχαίας προσπέλασης (DRAM) έως και 1TB για αποδοτικούς υπολογισμούς υψηλής απόδοσης. Ιδανικός για δίσκους στερεάς κατάστασης (SSD), σκληρούς δίσκους (HDD) και ενσωματωμένες λύσεις αποθήκευσης, καθώς και για εφαρμογές υπολογιστικής αποθήκευσης.

Cortex-A76

Η CPU Arm Cortex-A76 παρέχει επιδόσεις επιπέδου φορητού υπολογιστή με αποδοτικότητα smartphone, προσφέροντας την ίδια εμπειρία σε όλες τις κατηγορίες έξυπνων κινητών υπολογιστικών συσκευών. Παρέχει επιδόσεις φορητού υπολογιστή με αποδοτικότητα κινητού τηλεφώνου, φέρνοντας την ίδια εμπειρία κινητού τηλεφώνου (γρήγορη απόκριση, πάντα ενεργοποιημένη, πάντα συνδεδεμένη) σε όλες τις κατηγορίες έξυπνων κινητών υπολογιστικών

συσκευών. Με ανώτερη ενεργειακή απόδοση και πολύ μεγαλύτερη απόδοση σε ένα νήμα, η CPU Cortex-A76 παρατείνει τη διάρκεια ζωής της μπαταρίας και βελτιώνει την εμπειρία του χρήστη για συνεχή υψηλή απόδοση ακόμη και στις πιο σύνθετες υπολογιστικές εργασίες.

2.2.3 Πυρήνες

Ο πυρήνας ARM χρησιμοποιεί αρχιτεκτονική RISC. Η φιλοσοφία σχεδιασμού της αποσκοπεί στην παροχή απλών αλλά ισχυρών εντολών που εκτελούνται σε έναν μόνο κύκλο με υψηλή ταχύτητα ρολογιού. Η φιλοσοφία RISC επικεντρώνεται στη μείωση της πολυπλοκότητας των εντολών που εκτελούνται από το υλικό, επειδή είναι ευκολότερο να παρέχεται μεγαλύτερη ευελιξία και ευφυΐα στο λογισμικό παρά στο υλικό. Ως αποτέλεσμα, η σχεδίαση RISC έχει μεγαλύτερες απαιτήσεις από τον μεταγλωττιστή. Αντίθετα, ο παραδοσιακός υπολογιστής σύνθετου συνόλου εντολών (CISC) βασίζεται περισσότερο στο υλικό για τη λειτουργικότητα των εντολών και, κατά συνέπεια, οι εντολές CISC είναι πιο περίπλοκες. Παρόμοια με τις περισσότερες μηχανές RISC, ο ARM λειτουργεί με αρχιτεκτονική φόρτωσης-αποθήκευσης, οπότε μόνο οι εντολές φόρτωσης και αποθήκευσης εκτελούν πράξεις μνήμης και όλες οι άλλες αριθμητικές και λογικές πράξεις εκτελούνται μόνο στους καταχωρητές του επεξεργαστή. Στο σχήμα παρουσιάζεται το μοντέλο ροής δεδομένων του πυρήνα ARM.Παρόμοια με τις περισσότερες μηχανές RISC, ο ARM λειτουργεί με αρχιτεκτονική φόρτωσης-αποθήκευσης, οπότε μόνο οι εντολές φόρτωσης και αποθήκευσης εκτελούν πράξεις μνήμης και όλες οι άλλες αριθμητικές και λογικές πράξεις εκτελούνται μόνο στους καταχωρητές του επεξεργαστή. Στο σχήμα παρουσιάζεται το μοντέλο ροής δεδομένων του πυρήνα ARM. Στο σχήμα παρουσιάζεται το μοντέλο ροής δεδομένων του πυρήνα ARM. Στο οποίο ο πυρήνας ARM ως λειτουργικές μονάδες που συνδέονται με διαύλους δεδομένων,. Και τα βέλη αναπαριστούν τη ροή των δεδομένων, οι γραμμές αναπαριστούν τους διαύλους και τα κουτιά αναπαριστούν είτε μια λειτουργική μονάδα είτε μια περιοχή αποθήκευσης. Το σχήμα δείχνει όχι μόνο τη ροή των δεδομένων αλλά και τα αφηρημένα στοιχεία που συνθέτουν έναν πυρήνα ARM.

2.2.4 Πλεονεκτήματα

Υπάρχουν πολλαπλά πλεονεκτήματα ως προς τους λόγους για τους οποίους οι κατασκευαστές συσκευών θα επέλεγαν να εφαρμόσουν επεξεργαστές ARM στο προϊόν τους.

Χαμηλό κόστος

Οι επεξεργαστές ARM είναι οικονομικά προσιτοί στη δημιουργία και συνήθως δεν απαιτούν ακριβό εξοπλισμό για να το κάνουν. Οι επεξεργαστές ARM είναι συχνά ιδανικοί για συσκευές χαμηλότερου κόστους, όπως τα κινητά τηλέφωνα.

Απλός σχεδιασμός

Λόγω της σχεδίασης RISC, η οποία έχει λιγότερο περίπλοκη αρχιτεκτονική, οι επεξεργαστές ARM είναι απλούστεροι στη σχεδίαση και συχνά πολύ πιο συμπαγείς. Αυτό επιτρέπει την υλοποίηση των επεξεργαστών σε μικρότερες συσκευές. Αυτό αποτελεί πλεονέκτημα για τις αυξανόμενες απαιτήσεις των καταναλωτών για περισσότερες φορητές και φορητές συσκευές.

Απαιτήσεις χαμηλής ισχύος

Οι επεξεργαστές ARM λειτουργούν επίσης με χαμηλές απαιτήσεις ισχύος και καταναλώνουν λιγότερη ενέργεια σε σύγκριση με άλλους επεξεργαστές λόγω του σχεδιασμού της αρχιτεκτονικής RISC. Αυτό οφείλεται επίσης στην ικανότητά του να εκτελεί μόνο έναν κύκλο για την εκτέλεση μιας εντολής, μειώνοντας τις λειτουργίες.

Καλύτερη διάρκεια ζωής της μπαταρίας

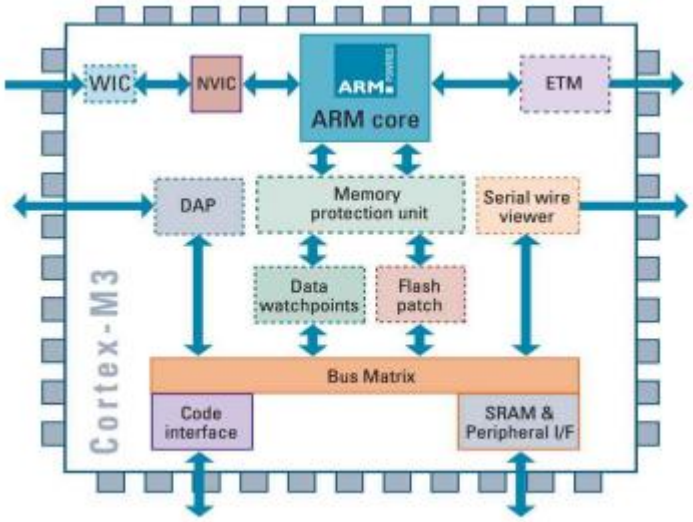
Επιπλέον, οι επεξεργαστές ARM καταναλώνουν λιγότερη μπαταρία λόγω του υπολογιστικού του συνόλου ενός κύκλου- ως εκ τούτου, οι επεξεργαστές ARM έχουν καλύτερη διάρκεια ζωής της μπαταρίας. Αυτό είναι επίσης συχνά ιδανικό για κινητές συσκευές που χρησιμοποιούνται συχνά χωρίς σύνδεση ρεύματος, σε αντίθεση με τους φορητούς υπολογιστές.

Χαμηλότερη παραγωγή θερμότητας

Οι επεξεργαστές ARM παράγουν επίσης λιγότερη θερμότητα, επιτρέποντας σε συσκευές όπως το smartphone ή το tablet να είναι λεπτότερες και να βρίσκονται συνεχώς στα χέρια του χρήστη.

2.2.5 Αρχιτεκτονική

Η αρχιτεκτονική ARM έχει εξελίχθηκε τα τελευταία χρόνια και έχει φθάσει σε επίπεδο τέτοιο ώστε να υποστηρίζει υλοποιήσεις ενός μμεγάλου φάσματος σημείων απόδοσης. Η αρχιτεκτονική απλότητα των επεξεργαστών ARM οδήγησαν σε υλοποιήσεις πολύ μικρού μεγέθους, και υλοποιήσεις που επιτρέπουν τη χρήση συσκευών με πολύ χαμηλή κατανάλωση ενέργειας . Το μέγεθος, η απόδοση και η πολύ χαμηλή κατανάλωση ενέργειας αποτελούν τα βασικά χαρακτηριστικά εξέλιξης της αρχιτεκτονικής ARM. Αυτός ο ARM είναι μια οικογένεια μικροελεκτών που αναπτύχθηκε από κατασκευαστές όπως η ST Microelectronics,η Motorola και ούτω καθεξής. Η αρχιτεκτονική ARM έρχεται με εντελώς διαφορετικές εκδόσεις όπως ARMv1, ARMv2, κ.λπ., και, κάθε μία έχει τα δικά της πλεονεκτήματα και μειονεκτήματα.



Επιπλέον η αρχιτεκτονική ARM περιέχει :

-Αριθμητική λογική μονάδα (ALU: Arithmetic Logic Unit)

Η ALU διαθέτει δύο εισόδους 32 bit. Η κύρια προέρχεται από το αρχείο καταχωρητών, ενώ η άλλη από τον μετατοπιστή. Οι σημαίες των καταχωρητών κατάστασης τροποποιούνται από τις εξόδους της ALU. Η έξοδος V-bit πηγαίνει στη σημαία V καθώς και η Count πηγαίνει στη σημαία C. Ενώ το πρωτεύον σημαντικότερο bit αντιπροσωπεύει πραγματικά τη σημαία S, η λειτουργία εξόδου της ALU γίνεται με NORed για να ληφθεί η σημαία Z. Η ALU διαθέτει δίαυλο λειτουργιών 4 bit που επιτρέπει την υλοποίηση έως και 16 opcode.

-Συντελεστής πολλαπλασιαστή Booth (Booth Multiplier Factor)

Ο παράγοντας πολλαπλασιαστή έχει 3 εισόδους 32 bit και οι είσοδοι επιστρέφουν από το αρχείο καταχωρητών. Η έξοδος του πολλαπλασιαστή είναι μόλις 32-Least Significant Bits του εμπορεύματος. Η αναπαράσταση οντότητας του παράγοντα πολλαπλασιαστή φαίνεται στο παραπάνω μπλοκ διάγραμμα. Ο πολλαπλασιασμός ξεκινά κάθε φορά που ενεργοποιείται η είσοδος αρχής 04. Το Fin της εξόδου γίνεται high όταν τελειώνει.

-Αλγόριθμος Booth (Booth Algorithm)

Ο αλγόριθμος Booth είναι ένας αξιοσημείωτος αλγοριθμικός κανόνας πολλαπλασιασμού για αριθμούς συμπληρώματος 2. Αντιμετωπίζει ομοιόμορφα τους θετικούς και τους αρνητικούς αριθμούς. Επιπλέον, οι διαδρομές των 0 ή 1 εντός του παράγοντα πολλαπλασιασμού παραλείπονται χωρίς να εκτελείται πρόσθεση ή αφαίρεση, δημιουργώντας έτσι δυνατό ταχύτερο πολλαπλασιασμό.

-Μετατοπιστής βαρελιού (Barrel Shifter)

Ο barrel shifter διαθέτει μια είσοδο 32-bit προς μετατόπιση. Αυτή η είσοδος επιστρέφει από το αρχείο καταχωρητών ή μπορεί να είναι άμεσα δεδομένα. Ο μετατοπιστής έχει διαφορετικές εισόδους ελέγχου που επιστρέφουν από τον καταχωρητή εντολών. Το πεδίο Shift εντός της εντολής ελέγχει τη λειτουργία του barrel shifter. Αυτό το πεδίο υποδεικνύει το είδος της μετατόπισης που πρέπει να εκτελεστεί (λογική αριστερά ή δεξιά, αριθμητική δεξιά ή περιστροφή δεξιά).

-Μονάδα ελέγχου (Control Unit)

Για κάθε μικροεπεξεργαστή, η μονάδα ελέγχου είναι η καρδιά της όλης διαδικασίας και είναι υπεύθυνη για τη λειτουργία του συστήματος, οπότε ο σχεδιασμός της μονάδας ελέγχου είναι το πιο σημαντικό μέρος στο πλαίσιο του όλου σχεδιασμού. Η μονάδα ελέγχου είναι μερικές φορές μια καθαρή σχεδίαση συνδυαστικού κυκλώματος. Εδώ, η μονάδα ελέγχου υλοποιείται με εύκολη μηχανή κατάστασης. Ο χρονισμός του επεξεργαστή περιλαμβάνεται επιπλέον στη μονάδα ελέγχου. Τα σήματα από τη μονάδα ελέγχου συνδέονται με κάθε συστατικό εντός του επεξεργαστή για την επίβλεψη της λειτουργίας του.

2.2.6 Χαρακτηριστικά

Έχουμε ήδη συζητήσει στην αρχή ότι οι επεξεργαστές ARM βασίζονται στην αρχιτεκτονική RISC. Έτσι, περιλαμβάνει βασικά χαρακτηριστικά της αρχιτεκτονικής RISC, τα οποία είναι τα εξής:

1. Οδηγίες(Instructions): Μια εντολή είναι 32 bit. Αυτό επιτρέπει τη λήψη κάθε εντολής σε έναν κύκλο, καθιστώντας έτσι τις λειτουργίες απλές. Εδώ κάθε εντολή έχει σταθερό μήκος, επιτρέποντας έτσι την ανάκτηση των μελλοντικών εντολών ενώ οι προηγούμενες εκτελούνται. Ενώ αυτή η δυνατότητα δεν παρέχεται από την αρχιτεκτονική CISC, καθώς στην περίπτωση αυτή, οι εντολές παρουσιάζουν μεταβλητό μέγεθος, η εκτέλεση των οποίων απαιτεί πολλούς κύκλους. Αυτό σημαίνει ότι οι επεξεργαστές ARM προσφέρουν απλή αποκωδικοποίηση εντολών.

2. Καταχωρητές(Registers): Οι μηχανές RISC περιέχουν μεγάλα ομοιόμορφα αρχεία καταχωρητών. Διαθέτει 37 καταχωρητές των 32 bit ο καθένας από τους οποίους μόνο 16 μπορούν να χρησιμοποιηθούν κάθε φορά. Σε αντίθεση με τους επεξεργαστές CISC όπου κάθε καταχωρητής είναι αφιερωμένος για συγκεκριμένους σκοπούς, στους RISC, κάθε καταχωρητής μπορεί να περιέχει είτε δεδομένα είτε διεύθυνση. Αυτό αυξάνει τη διαδικασία εκτέλεσης ολόκληρου του συστήματος.

3. Αγωγιμοποίηση(Pipelining): Βασίζεται σε διοχέτευση 3 σταδίων, η οποία παρέχει μέγιστη απόδοση. Βασικά, σε αυτή την περίπτωση, μέχρι να εκτελεστεί η πρώτη εντολή, η επόμενη θα αποκωδικοποιηθεί και η αμέσως επόμενη θα ληφθεί. Αυτό επιτρέπει την ταυτόχρονη λήψη, αποκωδικοποίηση και εκτέλεση. Αυτό σημαίνει ότι σε κάθε κύκλο υπάρχει η πρόοδος ενός βήματος που εξοικονομεί χρόνο και ως εκ τούτου για την εκτέλεση δεν απαιτούνται μικροκώδικες για την εκτέλεση εντολών όπως οι επεξεργαστές CISC.

4. Μοντέλο φόρτωσης/αποθήκευσης(Load/Store Model): Σε αυτή την αρχιτεκτονική, όλες οι πράξεις λαμβάνουν χώρα εντός του καταχωρητή. Μέσω της λειτουργίας φόρτωσης/αποθήκευσης, τα δεδομένα από τη μνήμη φορτώνονται στον καταχωρητή και πάνω σε αυτά τα δεδομένα εκτελείται η λειτουργία. Μόλις γίνει η πράξη, το αποτέλεσμα αποθηκεύεται στη μνήμη. Αυτό σημαίνει ότι, σε αντίθεση με τον CISC, ο οποίος υποστηρίζει την επεξεργασία δεδομένων στη μνήμη απευθείας, δεν παρέχει λειτουργίες που βασίζονται στη μνήμη.

2.2.7 Γιατί είναι τόσο πολύτιμη-μοναδική αρχιτεκτονική

Οι αρχιτεκτονικές ARM είναι ο πιο κοινός ηλεκτρονικός σχεδιασμός στον κόσμο, παρόλο που η x86 είναι πιο διαδεδομένη στην αγορά των διακομιστών. Οι αρχιτεκτονικές ARM χρησιμοποιούνται σχεδόν σε όλα τα σχέδια smartphone, καθώς και σε άλλες μικρές κινητές συσκευές και φορητούς υπολογιστές. Τα τσιπ x86 έχουν σχεδιαστεί για να βελτιστοποιούν τις επιδόσεις- οι επεξεργαστές που βασίζονται σε ARM έχουν σχεδιαστεί για να εξισορροπούν το κόστος με μικρότερα μεγέθη, χαμηλότερη κατανάλωση ενέργειας, χαμηλότερη παραγωγή θερμότητας, ταχύτητα και δυνητικά μεγαλύτερη διάρκεια ζωής της μπαταρίας.

Επειδή η Arm Holdings πωλάει σχέδια και όχι υλικό, αυτό επιτρέπει στους κατασκευαστές υλικού να προσαρμόζουν τη μικροαρχιτεκτονική στις συγκεκριμένες απαιτήσεις τους, διατηρώντας παράλληλα μικρό μέγεθος, υψηλές επιδόσεις και ενεργειακή απόδοση. Αυτό έχει τόσο πλεονεκτήματα όσο και μειονεκτήματα, καθώς σημαίνει επίσης ότι λειτουργικά συστήματα όπως το Linux, τα Windows και το Android πρέπει να υποστηρίζουν ένα ευρύτερο φάσμα υλικού.

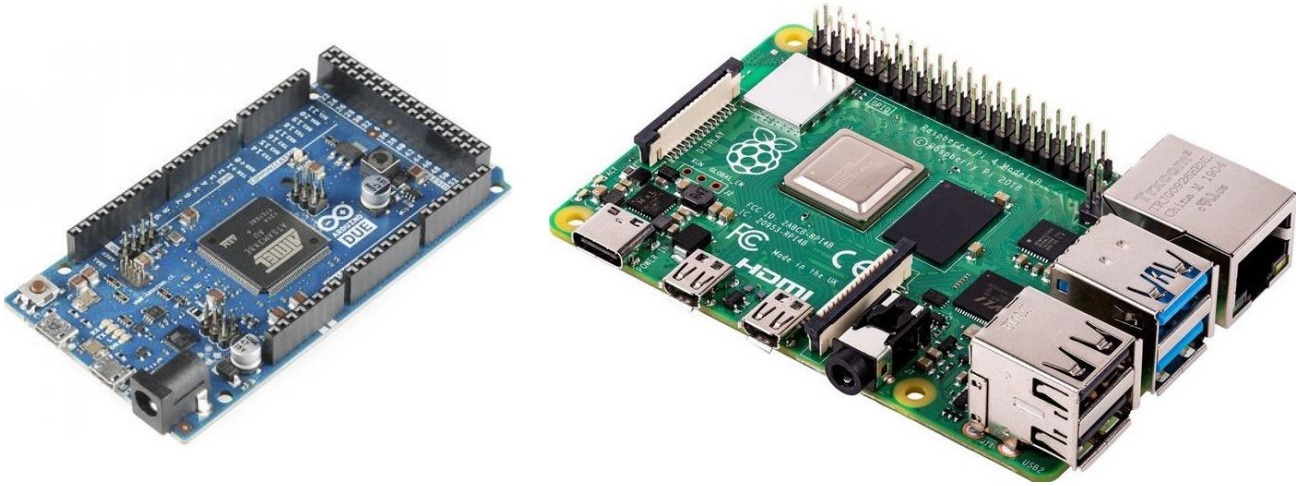
Αυτό δεν σημαίνει ότι οι αρχιτεκτονικές ARM χρησιμοποιούνται μόνο για μικρές κινητές συσκευές - ένας από τους ταχύτερους υπερυπολογιστές στον κόσμο, ο Fugaku που σχεδιάστηκε από τη Fujitsu και τη Riken, χρησιμοποιεί επεξεργαστή ARM. Στην περίπτωση της Fujitsu,

σχεδίασε το δικό της τσιπ ARM για τον υπερυπολογιστή της, αλλά η ARM παρέχει επίσης ένα σχεδιαστικό προφίλ για αρχιτεκτονικές HPC. Λόγω του μειωμένου μεγέθους, της χαμηλότερης κατανάλωσης ενέργειας και της χαμηλότερης θερμότητας (μειώνοντας την ανάγκη για πρόσθετη ψύξη), περισσότεροι οργανισμοί αρχίζουν να χρησιμοποιούν συστήματα ARM για τη δημιουργία κόμβων ή συστάδων για HPC και στόλους cloud (όπως το Amazon Web Services Graviton και το Microsoft Azure). Η αλυσίδα εργαλείων Arm Compiler for Linux είναι προσαρμοσμένη για την ανάπτυξη εφαρμογών HPC.

Η χρήση μιας αρχιτεκτονικής ARM δίνει στους σχεδιαστές υλικού περισσότερο έλεγχο στα σχέδια και τις επιδόσεις τους, καθώς και περισσότερο έλεγχο στις αλυσίδες εφοδιασμού τους. Αυτός ο συνδυασμός ελέγχου και επιδόσεων είναι ελκυστικός τόσο σε μικρές καταναλωτικές συσκευές όσο και σε υπολογιστικά περιβάλλοντα μεγάλης κλίμακας.

2.2.8 Πλακέτες-Συσκευές

Αναπτυξιακές Πλακέτες



Arduino Due R3: Το Arduino Due είναι μια πλακέτα μικροελεκτή που βασίζεται στη CPU Atmel SAM3X8E, η οποία είναι η πρώτη που βασίζεται σε 32-bit ARM core arduino.Το Arduino Due διαθέτει 54 ψηφιακές ακίδες εισόδου/εξόδου (εκ των οποίων οι 12 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM), 12 αναλογικές εισόδους, 4 UARTs (σειριακές θύρες υλικού), ένα ρολόι 84 MHz, μια σύνδεση με δυνατότητα USB-OTG, 2 DAC (ψηφιακά σε αναλογικά), 2 TWI, μια υποδοχή τροφοδοσίας, μια κεφαλή SPI, μια κεφαλή JTAG, ένα κουμπί επαναφοράς και ένα κουμπί διαγραφής.

Raspberry Pi 4 Model B: Το Raspberry Pi 4 Model B περιλαμβάνει τετραπύρηνο επεξεργαστή 64-bit υψηλής απόδοσης, υποστηρίζει διπλής οθόνης με ανάλυση έως 4K μέσω ενός ζεύγους θυρών micro-HDMI, αποκωδικοποίηση βίντεο υλικού έως 4Kp60, μνήμη RAM έως 4GB, ασύρματο LAN διπλής ζώνης 2,4/5,0 GHz, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, και δυνατότητα PoE (μέσω ενός ξεχωριστού πρόσθετου PoE HAT)

Συσκευές

- Ρομποτική (ηλεκτρομηχανικά συστήματα)

- Καταναλωτικά συστήματα πολυμέσων (ψηφιακές φωτογραφικές μηχανές κλπ.)

- Συστήματα πλοήγησης

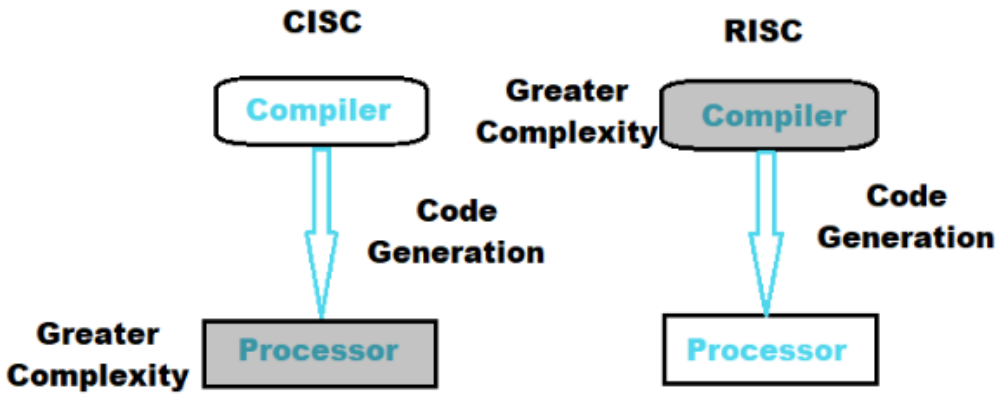
- Τηλεπικοινωνιακά συστήματα (κινητά τηλέφωνα κλπ.)

- Ηλεκτρονικά αεροσκαφών

2.2.9 Φιλοσοφία σχεδιασμού

Ο πυρήνας ARM χρησιμοποιεί αρχιτεκτονική RISC. Η RISC είναι μια φιλοσοφία σχεδιασμού που αποσκοπεί στην παροχή απλών αλλά ισχυρών εντολών που εκτελούνται μέσα σε έναν μόνο κύκλο σε υψηλή ταχύτητα ρολογιού. Η φιλοσοφία RISC επικεντρώνεται στη μείωση της πολυπλοκότητας των εντολών που εκτελούνται από το υλικό, επειδή είναι ευκολότερο να παρέχεται μεγαλύτερη ευελιξία και ευφυΐα στο λογισμικό παρά στο υλικό. Ως αποτέλεσμα, μια σχεδίαση RISC θέτει μεγαλύτερες απαιτήσεις στον μεταγλωττιστή. Αντίθετα, ο παραδοσιακός

υπολογιστής σύνθετου συνόλου εντολών (CISC) βασίζεται περισσότερο στο υλικό για τη λειτουργικότητα των εντολών και, κατά συνέπεια, οι εντολές CISC είναι πιο περίπλοκες. Το παρακάτω σχήμα απεικονίζει αυτές τις σημαντικές διαφορές.

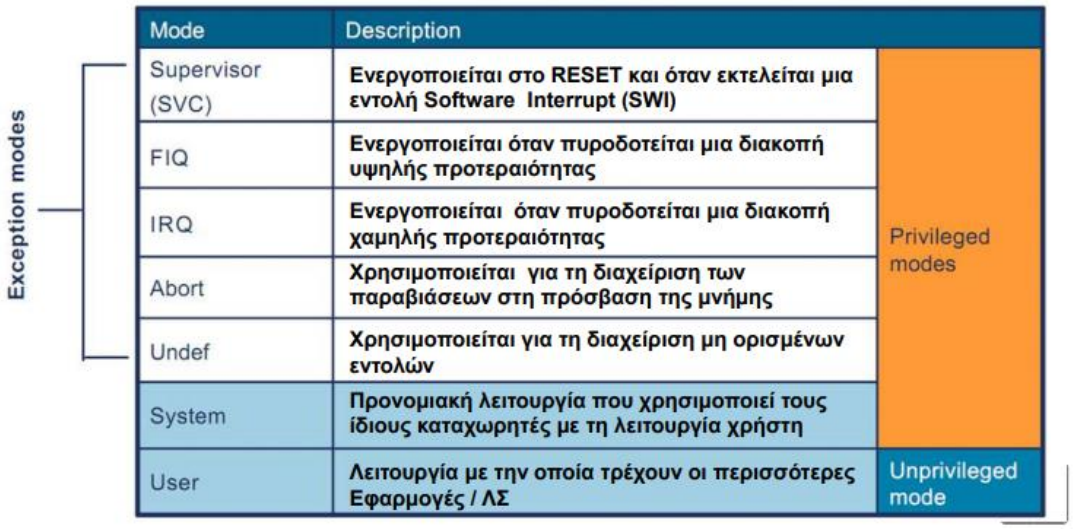


2.2.10 Λειτουργίες επεξεργαστή

Η αρχιτεκτονική ARM έχει επτά βασικές λειτουργίες:

-Κάθε λειτουργία έχει πρόσβαση στη δικιά της στοίβα και σε ένα

διαφορετικό σύνολο καταχωρητών.



2.3 Ο επεξεργαστής INTEL

Ιδρύθηκε στις 18 Ιουλίου 1968, από τους πρωτοπόρους των ημιαγωγών Gordon Moore (του νόμου του Moore) και Robert Noyce (1927-1990), και συνδέεται με την εκτελεστική ηγεσία και το όραμα του Andrew Grove. Η Intel αποτέλεσε βασικό συστατικό της ανόδου της Silicon Valley ως κέντρου υψηλής τεχνολογίας. Ο Noyce ήταν βασικός εφευρέτης του ολοκληρωμένου κυκλώματος (μικροτσίπ). Η Intel ήταν ένας πρώιμος προγραμματιστής τσιπ μνήμης SRAM και DRAM, τα οποία αντιπροσώπευαν την πλειονότητα των δραστηριοτήτων της μέχρι το 1981. Παρόλο που η Intel δημιούργησε το πρώτο εμπορικό τσιπ μικροεπεξεργαστή στον κόσμο το 1971, δεν ήταν μέχρι την επιτυχία του προσωπικού υπολογιστή (PC) που έγινε η κύρια δραστηριότητά της.

Η Intel είναι η παλαιότερη και πιο καθιερωμένη εταιρεία μικροεπεξεργαστών στον κόσμο, που παράγει τους πιο δημοφιλείς παγκοσμίως τσιπ μικροεπεξεργαστών. Αν και ίσως περισσότερο γνωστή για τους επεξεργαστές PC, η Intel χρησιμοποιούνται σε όλους σχεδόν τους τομείς της ηλεκτρονικής, συμπεριλαμβανομένης της αυτοκινητοβιομηχανίας, βιομηχανίας, αυτοματισμού, ρομποτικής, καταναλωτικών ηλεκτρονικά, επεξεργασία εικόνας, δικτύωση, κρυπτογράφηση, στρατός, κατασκευές, ιατρική, ενέργεια και άλλες βιομηχανίες. Οι σχεδιαστές που δεν είναι εξοικειωμένοι με την Intel αρχιτεκτονική μπορεί να έχουν ανησυχίες σχετικά με τις θεμελιώδεις έννοιες της αρχιτεκτονικής, την εσωτερική της λειτουργία ή την πολυπλοκότητά της.

2.3.1 Αρχιτεκτονική

Από το πρώτο μικροσκοπικό Intel 4004 τσιπ μικροεπεξεργαστή κατασκευάστηκε το 1971,η Intel παρήγαγε μια αδιάκοπη σειρά αναβαθμίσεων και βελτιώσεων του γνωστότερη οικογένεια μικροεπεξεργαστών στον κόσμο. Από τις πρώτες αρχές των 8-bit, η Intel αρχιτεκτονική της Intel περιλαμβάνει σήμερα μια σειρά μικροεπεξεργαστών 32-bit και 64-bit που απευθύνονται σε ένα ευρύ φάσμα εφαρμογών, απαιτήσεις επιδόσεων, επίπεδα ισχύος, και τιμές. Ο ακρογωνιαίος λίθος της αρχιτεκτονικής Intel είναι η συμβατότητά της. Κάθε νέα γενιά αρχιτεκτονικής Intel μικροεπεξεργαστή είναι ένα υπερσύνολο της των προκατόχων του, παρέχοντας προς τα πίσω συμβατότητα με παλαιότερα τσιπ και παλαιότερους λογισμικό, ενώ παράλληλα προσθέτει νέες ή βελτιωμένα χαρακτηριστικά. Αυτή η συμβατότητα επιτρέπει σε μηχανικούς, προγραμματιστές και ομάδες ανάπτυξης να επαναχρησιμοποιούν το λογισμικό και την ανάπτυξη λογισμικού από προηγούμενα έργα, προστατεύοντας τις επενδύσεις τους σε χρόνο και ταλέντο. Το καθιστά επίσης την ανάπτυξη νέων βασισμένων στην Intel συστήματα ευκολότερα, αξιοποιώντας τις δυνατότητες των προγραμματιστών εμπειρία των προγραμματιστών. Το λογισμικό μπορεί να επαναχρησιμοποιηθεί σε γενιές προϊόντων, και το προϊόν μπορούν να προστατεύσουν την επένδυσή τους (σε τόσο στο υλικό όσο και στο λογισμικό) με αποδοτικό τρόπο. Αν και η αρχική εργασία μπορεί να απαιτηθεί για την αξιοποίηση των πλεονεκτημάτων του νεότερα χαρακτηριστικά του μικροεπεξεργαστή, το παλιό λογισμικό θα εξακολουθεί να λειτουργεί ως έχει. Τα τσιπ αρχιτεκτονικής Intel έχουν προφανώς υποστεί πολλές αλλαγές κατά το παρελθόν 40+ χρόνια. Ένας κατάλογος των επί του παρόντος διαθέσιμων συσκευών είναι διαθέσιμη εδώ.1 Τα πρώτα τσιπ είχαν τεχνικούς αριθμούς εξαρτημάτων, όπως 8086, 80386 ή 80486. Αυτό οδήγησε σε στην κοινώς χρησιμοποιούμενη συντομογραφία "x86 αρχιτεκτονική", αναφερόμενος στην

τελευταία δύο ψηφία του αριθμού εξαρτήματος κάθε τσιπ. Από το 1993, η ονομασία "x86". έδωσε τη θέση της σε μια πιο αξιομνημόνευτη ονομασία (και προφορικά) ονόματα προϊόντων όπως Intel® Pentium® processor, Intel® Celeron®, επεξεργαστής Intel® Core™, και επεξεργαστής Intel® Atom™.Αν και κάθε κλάδος της ευρείας αρχιτεκτονικής Intel (ή x86) διατηρεί τα ίδια βασικά χαρακτηριστικά και λειτουργικότητα με τα προηγούμενα τσιπ, και διατηρεί την προς τα πίσω συμβατότητα με αυτά, κάθε νέα γενιά προσθέτει επίσης τα δικά της μοναδικά χαρακτηριστικά στο μείγμα.

2.3.2 Χαρακτηριστικά

Η μικροαρχιτεκτονική Core επέστρεψε σε χαμηλότερους ρυθμούς ρολογιού και βελτίωσε τη χρήση τόσο των διαθέσιμων κύκλων ρολογιού όσο και της ισχύος σε σύγκριση με την προηγούμενη μικροαρχιτεκτονική Net Burst των επεξεργαστών Pentium 4 και D.[4] Η μικροαρχιτεκτονική Core παρέχει αποδοτικότερα στάδια αποκωδικοποίησης, μονάδες εκτέλεσης, κρυφές μνήμες και διαύλους, μειώνοντας την κατανάλωση ισχύος των επεξεργαστών Core 2, ενώ παράλληλα αυξάνει την επεξεργαστική τους ικανότητα. Οι ΚΜΕ της Intel διέφεραν σε μεγάλο βαθμό ως προς την κατανάλωση ισχύος ανάλογα με το ρυθμό ρολογιού, την αρχιτεκτονική και τη διαδικασία ημιαγωγών, όπως φαίνεται στους πίνακες απορρόφησης ισχύος των ΚΜΕ.

Όπως και οι τελευταίες CPU Net Burst, οι επεξεργαστές με βάση τον Core διαθέτουν πολλαπλούς πυρήνες και υποστήριξη εικονικοποίησης υλικού. Ωστόσο, οι επεξεργαστές με βάση τον Core δεν διαθέτουν την τεχνολογία hyper-threading όπως στους επεξεργαστές Pentium 4. Αυτό οφείλεται στο γεγονός ότι η μικροαρχιτεκτονική Core βασίζεται στη μικροαρχιτεκτονική P6 που χρησιμοποιείται από τους Pentium Pro, II, III και M.

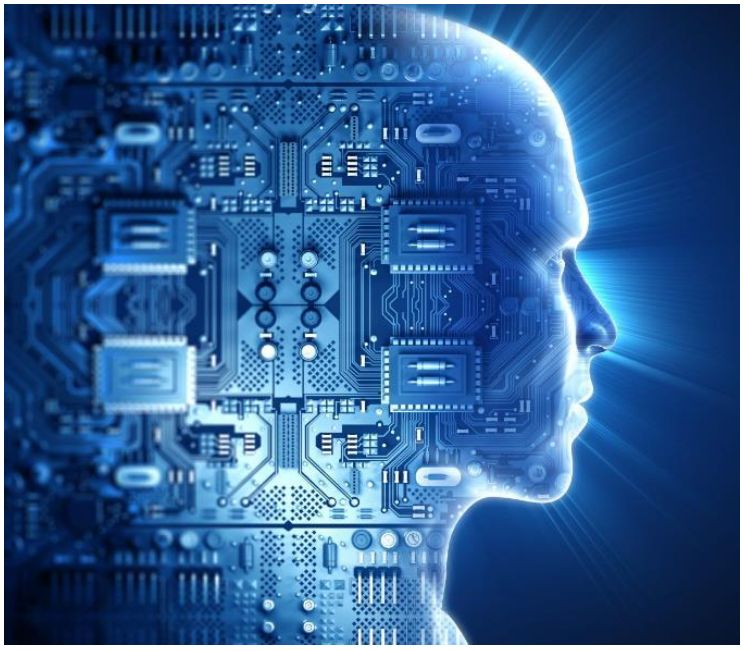
2.3.3 Τεχνολογίες

INTEL - MOBILEYE: ΑΥΤΌΝΟΜΗ ΟΔΉΓΗΣΗ

Η πορεία προς ένα αυτόνομο μέλλον αφορά πρωτίστως ασφαλέστερους δρόμους. Η Intel και η Mobileye θεωρούν ηθική επιταγή να παραδώσουν την τεχνολογία που θα καταστήσει αυτό το μέλλον εφικτό.

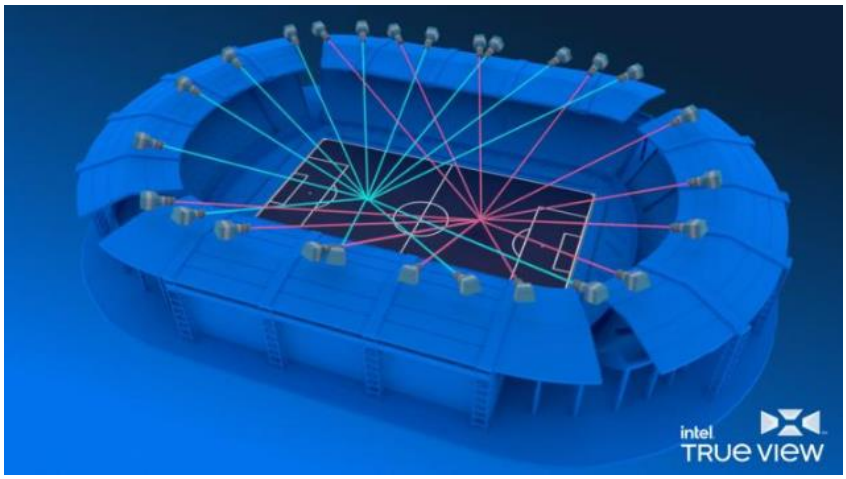
Αλλά δεν χρειάζεται να περιμένουμε τα πλήρως αυτόνομα οχήματα (AV) για να κάνουμε τους δρόμους μας ασφαλέστερους. Τα προηγμένα συστήματα υποβοήθησης οδηγού (ADAS) ή οι λύσεις υποβοηθούμενης οδήγησης μειώνουν ήδη τον αριθμό και τη σοβαρότητα των ατυχημάτων. Και όσο περισσότερα οχήματα μπορούμε να εξοπλίσουμε με αυτή την τεχνολογία, τόσο περισσότερο μπορούμε να προωθήσουμε αυτά τα οφέλη." Η Mobileye είναι ο ηγέτης στην υποβοηθούμενη οδήγηση σήμερα, παρούσα σε 12 από τα 16 αυτοκίνητα που έλαβαν βαθμολογία ασφάλειας 5 αστέρων από το EuroNCAP το 2018. Ήμασταν πρωτοπόροι στη χρήση της τεχνολογίας υπολογιστικής όρασης για τη διάσωση ζωών στο δρόμο. Σήμερα αυτή η τεχνολογία κλιμακώνεται για να γίνει τα δομικά στοιχεία για ένα πλήρως αυτόνομο όχημα. Ταυτόχρονα, η τεχνολογία που αναπτύσσουμε για τα AV αποδεικνύεται χρήσιμη για το επόμενο επίπεδο ADAS που ονομάζουμε επίπεδο 2+.

Intel - Τεχνητή νοημοσύνη



Η τεχνητή νοημοσύνη έχει εξαπολύσει μια νέα εποχή δημιουργικότητας και εφευρετικότητας. Σήμερα, οι τεχνολογίες της Intel τροφοδοτούν μερικές από τις πιο υποσχόμενες περιπτώσεις χρήσης της Τεχνητής Νοημοσύνης στις επιχειρήσεις, την κοινωνία και την έρευνα. Από το τεράστιο νέφος έως τη μικροσκοπική συσκευή, η Intel μετατρέπει την υπόσχεση ενός μετασχηματιστικού μοντέλου Τεχνητής Νοημοσύνης σε πραγματικότητα παγκόσμιας κλίμακας.

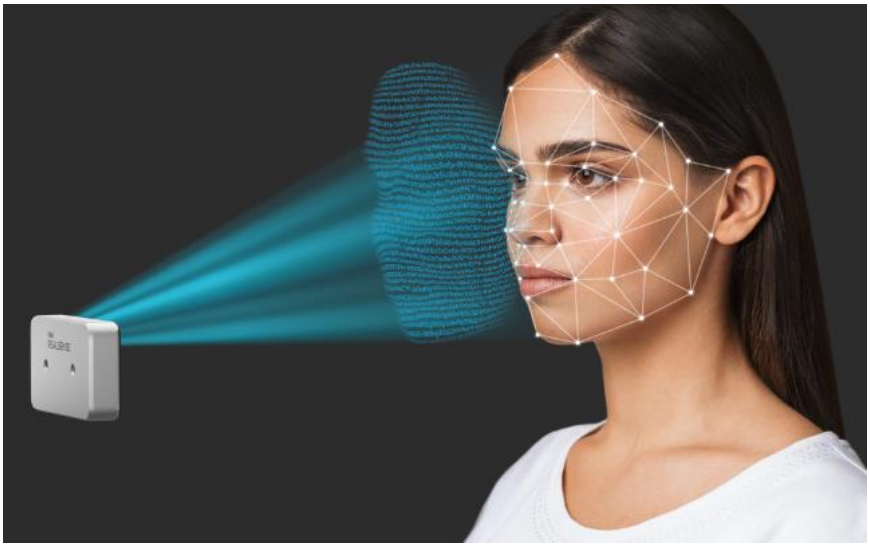
Intel – Αθλητισμός



Η Intel Sports επιτρέπει συναρπαστικές εμπειρίες πολυμέσων που πυροδοτούν το πάθος και τη γνώση σε όλη την αθλητική βιομηχανία. Αξιοποιώντας τη δύναμη του ογκομετρικού βίντεο, παρέχουμε στους συνεργάτες μας νέες εμπειρίες για την εμπλοκή των φιλάθλων μέσω της δια δραστικότητας, της εξατομίκευσης και των απεριόριστων προοπτικών του παιχνιδιού.

Το True View είναι η πλατφόρμα ογκομετρικού βίντεο της Intel Sports για τη λήψη, επεξεργασία και παραγωγή δεδομένων. Αυτή η νέα μορφή πολυμέσων επιτρέπει άπειρη αφήγηση ιστοριών από μία λήψη. Χρησιμοποιώντας την τεχνολογία ογκομετρικής καταγραφής, το υλικό καταγράφεται από δεκάδες κάμερες 5K Ultra HD για τη δημιουργία ενός εικονικού περιβάλλοντος σε θεαματικό, πολυπροοπτικό 3D, επιτρέποντας στους φιλάθλους να βιώσουν μια στιγμή στο γήπεδο, το γήπεδο ή τον αγωνιστικό χώρο από οποιαδήποτε γωνία.

Αυθεντικοποίηση προσώπου Intel RealSense ID



Η Intel παρουσίασε το Intel RealSense ID, μια λύση για τη συσκευή που συνδυάζει έναν ενεργό αισθητήρα βάθους με ένα εξειδικευμένο νευρωνικό δίκτυο, σχεδιασμένο για να παρέχει ασφαλή, ακριβή και συνειδητοποιημένο από τον χρήστη έλεγχο ταυτότητας προσώπου. Το Intel RealSense ID λειτουργεί με έξυπνες κλειδαριές, έλεγχο πρόσβασης, σημεία πώλησης, ATM, περίπτερα και άλλα.

2.3.4 Οικογένειες Intel επεξεργαστών

- Intel Atom

- Intel Celeron

- Intel Core

- Intel Pentium

- Intel Xeon

2.3.5 Πλεονεκτήματα

Όταν αποφασίζετε για έναν επεξεργαστή για την τροφοδοσία μιας συσκευής προσωπικού υπολογιστή, λίγα ονόματα έχουν τη φήμη της Intel Corporation, με έδρα την Καλιφόρνια. Η Intel είναι ηγέτης της αγοράς και γνωστό όνομα εδώ και δεκαετίες με μια ποικιλία τσιπ επεξεργαστών, όπως τα Core, Xeon και Pentium. Μάρκες όπως η Apple, η Lenovo και η Dell χρησιμοποιούν τσιπ της Intel στους υπολογιστές τους εδώ και χρόνια. Η Intel έχει κερδίσει τη φήμη της με την πάροδο των ετών, οπότε οι επεξεργαστές της Intel έχουν σίγουρα το μερίδιο των πλεονεκτημάτων τους. Ας σταθμίσουμε μερικά από τα πλεονεκτήματα των επεξεργαστών Intel.

- **Συμβατότητα**: Οι επεξεργαστές Intel είναι ευρέως διαθέσιμοι. Αυτό παρέχει στους καταναλωτές μια πιο ευέλικτη δυνατότητα επιλογών Η/Υ, μεγαλύτερη ποικιλία χαρακτηριστικών και μπορεί να βρεθεί σε συσκευές που διατίθενται σε χαμηλότερη τιμή.

- **Κατανάλωση ενέργειας:** Οι επεξεργαστές Intel στους φορητούς υπολογιστές δεν είναι ενεργοβόροι, καταναλώνοντας λιγότερη ενέργεια από άλλες CPU της αγοράς. Για παράδειγμα, η CPU της AMD χρησιμοποιεί έως και διπλάσια ή και τριπλάσια ισχύ σε σχέση με την Atom CPU της Intel, η οποία τροφοδοτεί τα Netbook. Η διαφορά γίνεται εύκολα αντιληπτή στην μπαταρία σε σύγκριση.

- **Παραγωγή θερμότητας:** Λόγω είναι χαμηλότερη η κατανάλωση ισχύος, τα τσιπ επεξεργασίας της Intel παράγουν λιγότερη θερμότητα. Σε έναν μίνι υπολογιστή, οι κατασκευαστές συσκευάζουν τα εξαρτήματα πιο σφιχτά μεταξύ τους, οπότε η λιγότερη θερμότητα που παράγεται από τον επεξεργαστή έχει ως αποτέλεσμα όλα τα υπόλοιπα να παραμένουν πιο δροσερά.

- **Παραγωγική ικανότητα:** Με 15 εργοστάσια ανάπτυξης επεξεργαστών σε παγκόσμιο επίπεδο, η Intel μπορεί να παράγει μεγαλύτερο όγκο παραγωγής, πράγμα που σημαίνει ότι οι επεξεργαστές της θα είναι ευρύτερα διαθέσιμοι και θα χρησιμοποιούνται σε σημαντικό αριθμό μηχανημάτων.

- **Ενσωματωμένα γραφικά:** Σε γενικές γραμμές, όταν εξετάζετε μια CPU, καταλήγετε είτε στην AMD είτε στην Intel. Ένα πράγμα που δίνει στην Intel το πάνω χέρι είναι ότι έρχεται με ενσωματωμένα γραφικά Iris ή HD από το κουτί. Αυτό σημαίνει ότι μπορείτε να περάσετε κατευθείαν στο gaming ή στη ροή βίντεο υψηλής ποιότητας χωρίς GPU.

2.4 Ο επεξεργαστής Atmel

Η Atmel Corporation ήταν δημιουργός και κατασκευαστής ημιαγωγών πριν απορριφθεί από την Microchip Technology το 2016. Η Atmel ιδρύθηκε το 1984. Η εταιρεία επικεντρώθηκε σε ενσωματωμένα συστήματα που βασίζονται σε μικροελεκτές. Τα προϊόντα της περιλάμβαναν μικροελεκτές (8-bit AVR, 32-bit AVR, 32-bit ARM-based, automotive grade, και 8-bit Intel 8051 derivatives) συσκευές ραδιοσυχνοτήτων (RF), συμπεριλαμβανομένων των Wi-Fi, EEPROM και flash memory devices, συμμετρικά και ασύμμετρα security chips, αισθητήρες και ελεγκτές αφής και προϊόντα ειδικών εφαρμογών. Η Atmel παρέχει τις συσκευές της ως τυποποιημένα προϊόντα, ολοκληρωμένα κυκλώματα ειδικών εφαρμογών (ASIC) ή τυποποιημένα προϊόντα ειδικών εφαρμογών (ASSP) ανάλογα με τις απαιτήσεις των πελατών της. Η Atmel εξυπηρετεί εφαρμογές όπως καταναλωτικές, επικοινωνίες, δίκτυα υπολογιστών, βιομηχανικές, ιατρικές, αυτοκινητοβιομηχανίες, αεροδιαστημικές και στρατιωτικές εφαρμογές. Ειδικεύεται στους μικροελεγκτές και τα συστήματα αφής, ιδίως για ενσωματωμένα συστήματα.

2.4.1 Οικογένειες επεξεργαστών

ATMEL SAM C20:

Οι συσκευές SMART SAM C λειτουργούν σε μέγιστη συχνότητα 48MHz και φτάνουν τα 2,46 Coremark/MHz. Έχουν σχεδιαστεί για απλή και διαισθητική μετάβαση με πανομοιότυπες περιφερειακές μονάδες, δεκαεξαδικό συμβατό κώδικα, πανομοιότυπο γραμμικό χάρτη διευθύνσεων και pin συμβατά μονοπάτια μετάβασης μεταξύ όλων των συσκευών της σειράς προϊόντων. Όλες οι συσκευές περιλαμβάνουν έξυπνες και ευέλικτες περιφερειακές μονάδες, σύστημα συμβάντων Atmel Event System για διαπεριφερειακή σηματοδότηση και υποστήριξη για χωρητικές διεπαφές χρήστη με κουμπιά αφής, ρυθμιστικά και τροχούς.

ATMEL SAM D20:

Οι μικροελεγκτές (MCU) SAM D είναι πραγματικά διαφοροποιημένοι μικροελεγκτές γενικής χρήσης που είναι ιδανικοί για πολλές βιομηχανικές και καταναλωτικές εφαρμογές χαμηλής κατανάλωσης ενέργειας και ευαίσθητες στο κόστος. Αυτές οι MCUs προσφέρουν εξαιρετικές επιλογές διασύνδεσης και περιφερειακών με εντυπωσιακή απόδοση χαμηλής κατανάλωσης ενέργειας. Με τη σειρά μικροελεκτών (MCU) Atmel SAM D20 με βάση τον ARM Cortex-M0+ η Atmel θέτει ένα νέο σημείο αναφοράς για την ευελιξία και την ευκολία χρήσης. Ο SAM D20 συνδυάζει τις επιδόσεις και την ενεργειακή απόδοση μιας MCU με βάση τον ARM Cortex-M0+ με μια βελτιστοποιημένη αρχιτεκτονική και ένα σύνολο περιφερειακών, δίνοντάς σας έναν πραγματικά διαφοροποιημένο μικροελεκτή γενικής χρήσης, ιδανικά κατάλληλο για βιομηχανικές και καταναλωτικές εφαρμογές χαμηλής κατανάλωσης και ευαίσθητες στο κόστος.

ATMEL SAM 4L:

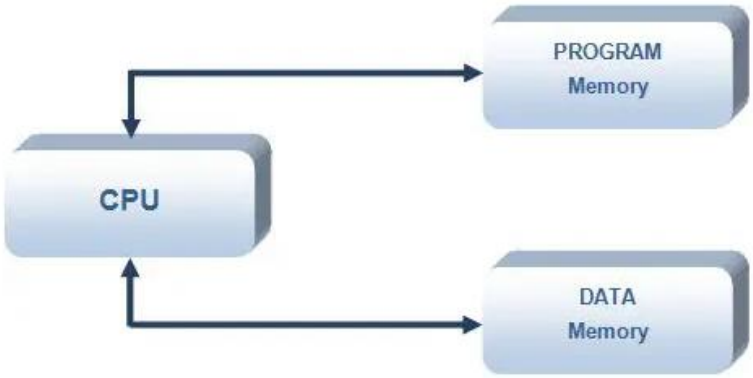
Η οικογένεια μικροελεκτών (MCUs) SAM L είναι κατασκευασμένη με την καινοτόμο τεχνολογία picoPower για να παρέχει την καλύτερη στην κατηγορία της χαμηλή κατανάλωση ενέργειας έως 25 μA/MHz σε ενεργό λειτουργία, κάτω από 100 nA σε κατάσταση αναστολής λειτουργίας και γρήγορους χρόνους αφύπνισης 1,2 uS. Αυτές οι MCUs έχουν επιτύχει βαθμολογία 410 στο ULPMark, πιστοποιημένη από το EEMBC, η οποία είναι η υψηλότερη βαθμολογία για μια συσκευή κατηγορίας Arm Cortex-M23 ή Cortex-M0+. Εκτός από τις δυνατότητες εξαιρετικά χαμηλής κατανάλωσης ενέργειας, αυτές οι συσκευές διαθέτουν βελτιωμένο Peripheral Touch Controller (PTC), ισχυρή ασφάλεια, τεχνολογία Arm TrustZone, κρυπτογράφηση Advanced Encryption Standard (AES), Full-Speed USB host και device, Event System και SleepWalking, 12-bit analog, ενσωματωμένους op amps και πολλά άλλα.

2.4.2 Αρχιτεκτονική

Οι μικροελεγκτές AVR βασίζονται στην προηγμένη αρχιτεκτονική RISC και αποτελούνται από 32 x 8-bit καταχωρητές εργασίας γενικού σκοπού. Μέσα σε έναν μόνο κύκλο ρολογιού, ο AVR μπορεί να λάβει εισόδους από δύο καταχωρητές γενικής χρήσης και να τις μεταφέρει στην ALU για την εκτέλεση της ζητούμενης λειτουργίας και να μεταφέρει το αποτέλεσμα σε έναν αυθαίρετο καταχωρητή. Η ALU μπορεί να εκτελέσει αριθμητικές καθώς και λογικές πράξεις.

στις εισόδους από τον καταχωρητή ή μεταξύ του καταχωρητή και μιας σταθεράς. Πράξεις ενός καταχωρητή, όπως η λήψη συμπληρώματος, μπορούν επίσης να εκτελεστούν στην ALU. Βλέπουμε ότι ο AVR δεν διαθέτει κανένα καταχωρητή όπως ο συσσωρευτής όπως στην οικογένεια μικροελεκτών 8051- οι πράξεις μπορούν να εκτελεστούν μεταξύ οποιουδήποτε από τους καταχωρητές και μπορούν να αποθηκευτούν σε οποιονδήποτε από αυτούς.

Ο AVR ακολουθεί τη μορφή αρχιτεκτονικής Harvard στην οποία ο επεξεργαστής είναι εξοπλισμένος με ξεχωριστές μνήμες και διαύλους για το πρόγραμμα και τις πληροφορίες δεδομένων. Εδώ, ενώ εκτελείται μια εντολή, η επόμενη εντολή προεπιλέγεται από τη μνήμη προγράμματος.



2.4.3 Εκτέλεση προγράμματος

Όπως πιθανώς γνωρίζετε, ένα πρόγραμμα μετατρέπεται πάντα σε κώδικα συναρμολόγησης/μηχανής, ο οποίος δεν είναι τίποτα άλλο παρά ένα μάτσο δυαδικοί αριθμοί, ανεξάρτητα από τη γλώσσα προγραμματισμού που χρησιμοποιείτε. Έτσι, ο κώδικας μηχανής φορτώνεται στη μνήμη flash της MCU σε μια συγκεκριμένη θέση (όχι απαραίτητα στο $0000, εξαρτάται από κάποιες ρυθμίσεις για τις οποίες θα μιλήσω αργότερα). Μετά από αυτό, ο H/Y ρυθμίζεται στη διεύθυνση έναρξης του προγράμματος & αρχίζει η εκτέλεσή του. Το PC αυξάνεται κατά 1 μετά την ολοκλήρωση της προηγούμενης εντολής (στην πραγματικότητα το ποσό της αύξησης εξαρτάται και από την εντολή, αλλά προς το παρόν, ξεχάστε το αυτό). Ένα άλλο πολύ σημαντικό εξάρτημα μέσα στην MCU είναι μια γεννήτρια παλμών ή ένα "ρολόι" το οποίο απλά παράγει τετραγωνικούς παλμούς σε σταθερή συχνότητα. Ο σκοπός του ρολογιού είναι να δημιουργήσει έναν ρυθμό μέσα στην MCU, ώστε να μπορεί να βεβαιωθεί για την ταυτόχρονη επεξεργασία κάθε μονάδας.

2.4.4 Ανάπτυξη

Η Atmel ιδρύθηκε το 1984. Το όνομα σημαίνει "προηγμένη τεχνολογία για μνήμη και λογική", αν και αρχικά η εστίαση ήταν στη μνήμη. Ο Γιώργος Περλέγκος, ο ιδρυτής, είχε εργαστεί στην ομάδα μνήμης της Intel, όταν η Intel ήταν εταιρεία μνήμης και όχι μικροεπεξεργαστών, αν και αυτό δεν εμπόδισε την Intel να τους μηνύσει μετά από μερικά χρόνια για παραβίαση πατέντας. Αρχικά η Atmel δεν είχε εργοστάσιο, αλλά αγόρασε το παλιό εργοστάσιο της Honeywell στο Κολοράντο Σπρινγκς, συγκεντρώνοντας κάποια επιχειρηματικά κεφάλαια για να το κάνει. Επέκτειναν το εργοστάσιο μετά την αγορά του κατασκευαστή FPGA Concurrent Logic το 1991, τη χρονιά που μπήκαν στο χρηματιστήριο.

Στη συνέχεια ανέπτυξαν τον πρώτο τους μικροελεκτή βασισμένο στον Intel 8051 αν και με πολύ υψηλότερες επιδόσεις. Εξακολουθούν να έχουν τον 8051 στον κατάλογο των προϊόντων τους σήμερα, τόσο ως αυτόνομο μικροελεκτή όσο και ως πυρήνα για να μπορούν να τον ενσωματώσουν

σε πιο προσαρμοσμένες συσκευές. Στα μέσα της δεκαετίας του 1990 απέκτησαν άδεια χρήσης της αρχιτεκτονικής ARM και σήμερα διαθέτουν ένα ευρύ χαρτοφυλάκιο συσκευών με βάση την ARM.Ανέπτυξαν επίσης τη σειρά μικροελεκτών AVR που ήταν οι πρώτοι που χρησιμοποίησαν μνήμη flash. Το όνομα προέρχεται από τα αρχικά των Νορβηγών προγραμματιστών του αρχικού πυρήνα. Σήμερα, το έργο Arduino ανοιχτού κώδικα είναι χτισμένο γύρω από τον AVR και απευθύνεται όχι σε επαγγελματίες πληροφορικής, αλλά σε χομπίστες, σχεδιαστές και κατασκευαστές.

2.4.5 Χαρακτηριστικά

- Πολυλειτουργικό
- Πολλαπλοί εσωτερικοί ταλαντωτές
- Μνήμη flash έως 256 KB
- Εσωτερική μνήμη δεδομένων EEPROM έως 4 KB
- Εσωτερική μνήμη SRAM έως 16 KB
- Δύο χρονοδιακόπτες 8 bit και ένας χρονοδιακόπτης 16 bit

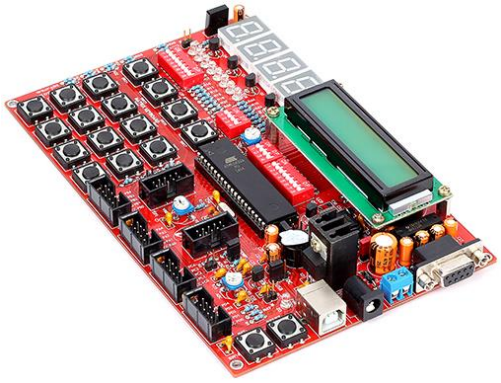
2.4.6 Η σημασία

Οι μικροελεγκτές (MCU) AVR της Atmel προσφέρουν ισχυρές επιδόσεις, χαμηλή κατανάλωση ενέργειας, υψηλές ταχύτητες, συνδεσιμότητα και υψηλή ενσωμάτωση συστήματος. Βασισμένοι σε έναν κινητήρα RISC ενός κύκλου που συνδυάζει ένα πλούσιο σύνολο εντολών, οι AVR MCUs παρέχουν σχεδόν 1 MIPS (εκατομμύρια εντολές ανά δευτερόλεπτο) ανά megahertz και είναι βελτιστοποιημένοι για ελάχιστο μέγεθος κώδικα και μέγιστη υπολογιστική απόδοση. Είναι ιδανικές για ένα ευρύ φάσμα εφαρμογών, όπως βιομηχανικός έλεγχος, ασύρματες, ιατρικές και κοινωφελείς μετρήσεις, πύλες επικοινωνίας, έλεγχος αισθητήρων, συσκευές (λευκές συσκευές, μικρές συσκευές κουζίνας, προσωπική υγειονομική περίθαλψη κ.λπ.) και φορητά προϊόντα που λειτουργούν με μπαταρίες. Επιπλέον, οι AVR MCU σας επιτρέπουν να διαθέσετε τα προϊόντα σας στην αγορά γρήγορα.

Είτε εργάζεστε σε νέα, υφιστάμενα ή παλαιότερα σχέδια, η ευρεία γκάμα συσκευών AVR της Atmel παρέχει τα πιο πρόσφατα χαρακτηριστικά και λειτουργίες. Αυτές οι συσκευές παρέχουν επίσης υψηλή ολοκλήρωση, τη χαμηλότερη κατανάλωση ενέργειας, ένα ολοκληρωμένο σύνολο ισχυρών ενσωματωμένων περιφερειακών και ευκολία χρήσης παγκόσμιας κλάσης.

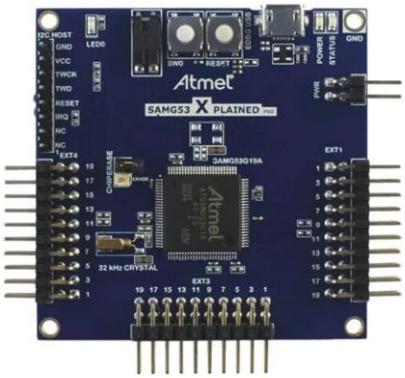
2.4.7 Εργαλεία ανάπτυξης και Kit αξιολόγησης

Πλακέτα ανάπτυξης AVR ATmega32



Η πλακέτα ανάπτυξης AVR ATmega32 μπορεί να χρησιμοποιηθεί για την αξιολόγηση και την επίδειξη των δυνατοτήτων του μικροελεκτή AVR ATmega32. Η υποδοχή MCU στην πλακέτα παρέχει υποστήριξη για τη συσκευασία 40 pin DIP του ελεγκτή AVR ATmega32. Η πλακέτα έχει σχεδιαστεί για εφαρμογές γενικού σκοπού και περιλαμβάνει μια ποικιλία υλικού για την άσκηση των περιφερειακών του μικροελεκτή. Είναι ένα φανταστικό εργαλείο για την αποσφαλμάτωση κώδικα, την ανάπτυξη και την κατασκευή πρωτοτύπων. Ο ολοκαίνουργιος σχεδιασμός της πλακέτας την έχει καταστήσει πολύ πιο φιλική προς το χρήστη από τους προκατόχους της, ιδανική για σκοπούς εκπαίδευσης και ανάπτυξης. Τα εξαρτήματα που χρησιμοποιούνται σε αυτή την πλακέτα είναι υψηλής ποιότητας και η πλακέτα είναι υψηλής ποιότητας PCB δύο στρώσεων PTH PCB, γεγονός που καθιστά αυτή την πλακέτα ιδιαίτερα ανθεκτική. Όταν αρχίσετε να χρησιμοποιείτε την αναπτυξιακή πλακέτα AVR ATmega32 θα διαπιστώσετε ότι είναι ένα από τα καλύτερα εργαλεία ανάπτυξης που διατίθενται για τον ελεγκτή AVR ATmega32. Παρακάτω θα βρείτε τα βασικά χαρακτηριστικά αυτού του προϊόντος.

Atmel Microcontroller Development Kit ATSAMG53-XPRO



Το kit αξιολόγησης Atmel SAM G53 Explained Pro επιτρέπει στους μηχανικούς σχεδιασμού να δημιουργήσουν πρωτότυπα και να αξιολογήσουν μικροελεγκτές με επεξεργαστή SAMG53 Cortex

M4. Το kit αξιολόγησης Explained Pro MCU διαθέτει ενσωματωμένο αποσφαλματωτή και δεν απαιτούνται εργαλεία για τον προγραμματισμό ή την αποσφαλμάτωση της συσκευής ATSAMG53N19. Το Atmel Studio IDP σας βοηθάει στην ανάπτυξη ή αποσφαλμάτωση των ενσωματωμένων συστημάτων των εφαρμογών ARM Cortex-M της Atmel.

UNO-R3 MEGA2560 v CH340

Uno R3 - συμβατό με την πλακέτα Arduino Uno R3. Βασίζεται στον μικροελεκτή Atmel ATmega328 και στο τσιπ διασύνδεσης USB-UART CH340G. Πλακέτα για λειτουργικότητα παρόμοια με το Arduino Uno (ATmega 168/328). Είναι μια οικονομική, αλλά η ίδια σταθερή, και χρησιμοποιεί τα πρωτότυπα τσιπ Atmel ATmega (16 MHz). Η πλακέτα χρησιμοποιεί το τσιπ CH340G ως μετατροπέα UART-USB. Chip CH340G - είναι μια οικονομική λύση. Όταν εργάζεστε στη συχνότητα 12Mhz, δίνοντας ένα σταθερό αποτέλεσμα της ανταλλαγής δεδομένων (πρέπει να εγκαταστήσετε τους οδηγούς στον υπολογιστή). Uno R3 - συνδέεται με τον υπολογιστή μέσω καλωδίου micro USB (χρησιμοποιείται για σχεδόν όλα τα smartphones Android).

2.5 Ο επεξεργαστής Motorola

Η Motorola Inc. είναι ένας από τους κορυφαίους κατασκευαστές μικροεπεξεργαστών. Ένα ολοκληρωμένο κύκλωμα που περιέχει ολόκληρη την κεντρική μονάδα επεξεργασίας ενός υπολογιστή σε ένα μόνο τσιπ. Μέχρι τις αρχές της δεκαετίας του 1990, οι μικροεπεξεργαστές Motorola χρησιμοποιούνταν σε όλους τους υπολογιστές Apple Macintosh και σε πολλούς σταθμούς εργασίας. Ωστόσο, μετά την ανάπτυξη του τσιπ 68040 το 1989, η Motorola άλλαξε την εστίασή της από τη σειρά τσιπ CISC 680×0 σε τεχνολογίες RISC. Το 1993, η Motorola συμμετείχε μαζί με την Apple Computer και την IBM στο σχεδιασμό μιας νέας αρχιτεκτονικής RISC που θα αποτελούσε τη βάση της επόμενης γενιάς προσωπικών υπολογιστών. Η προσπάθεια αυτή κορυφώθηκε με την παρουσίαση της αρχιτεκτονικής PowerPC το 1994.

2.5.1 Η ανάπτυξη του επεξεργαστή

Η πρώτη ευρέως παραγόμενη CPU της Motorola ήταν η Motorola 6800. Παρόλο που ήταν μια ικανή κατασκευή, επισκιάστηκε από πιο ισχυρά σχέδια, όπως το Zilog Z80, και λιγότερο ακριβά

σχέδια, όπως το MOS Technology 6502. Καθώς οι προοπτικές πωλήσεων του 6800 εξασθένησαν, η Motorola ξεκίνησε μια νέα σχεδίαση για να τον αντικαταστήσει. Αυτό έγινε το έργο Motorola Advanced Computer System on Silicon (MACSS), που ξεκίνησε το 1976 υπό τη διεύθυνση του Tom Gunter, του κύριου αρχιτέκτονα του. Στόχος του MACSS ήταν η ανάπτυξη μιας εντελώς νέας αρχιτεκτονικής χωρίς συμβατότητα προς τα πίσω με τον 6800. Τελικά, διατήρησε μια λειτουργία συμβατότητας πρωτοκόλλου διαύλου για τις υπάρχουσες περιφερειακές συσκευές του 6800 και παρήχθη μια έκδοση με δίαυλο δεδομένων 8 bit. Ωστόσο, οι σχεδιαστές επικεντρώθηκαν κυρίως στο μέλλον ή στην προς τα εμπρός συμβατότητα, η οποία δίνει στο σχεδιασμό του 68000 ένα προβάδισμα έναντι μεταγενέστερων αρχιτεκτονικών συνόλου εντολών (ISA) των 32 bit. Για παράδειγμα, οι καταχωρητές της CPU έχουν πλάτος 32 bit, αν και λίγες αυτοτελείς δομές στον ίδιο τον επεξεργαστή λειτουργούν σε 32 bit κάθε φορά. Η ομάδα MACSS βασίστηκε σε μμεγάλο βαθμό στην επιρροή του σχεδιασμού επεξεργαστών μικροϋπολογιστών, όπως τα συστήματα PDP-11 και VAX, τα οποία βασίζονται παρομοίως σε μικροκώδικες.

Στα μέσα της δεκαετίας του 1970, οι κατασκευαστές μικροεπεξεργαστών 8-bit έκαναν αγώνα δρόμου για να παρουσιάσουν τη γενιά των 16-bit. Η National Semiconductor ήταν πρώτη με τους επεξεργαστές IMP-16 και PACE το 1973 και το 1974 αντίστοιχα, αλλά αυτοί είχαν προβλήματα με την ταχύτητα. Η Intel δούλευε πάνω στον προηγμένο επεξεργαστή 16/32-bit Intel iAPX 432 (γνωστός και ως 8800) από το 1975 και στον Intel 8086 από το 1976 (παρουσιάστηκε το 1978 αλλά διαδόθηκε ευρέως με τη μορφή του σχεδόν πανομοιότυπου 8088 στον IBM PC λίγα χρόνια αργότερα). Η καθυστερημένη άφιξη στην αρένα των 16-bit προσφέρει στον νέο επεξεργαστή περισσότερα τρανζίστορ (περίπου 40.000[αναφορά που απαιτείται] ενεργά έναντι 20.000 ενεργών στον 8086), μακροεντολές 32-bit και αναγνωρισμένη γενική ευκολία χρήσης.

2.5.2 Παραλλαγές

Το 1982, ο 68000 έλαβε μια μικρή αναβάθμιση της αρχιτεκτονικής του συνόλου εντολών (ISA) για να υποστηρίξει εικονική μνήμη και να συμμορφωθεί με τις απαιτήσεις εικονικοποίησης των Popek και Goldberg. Το ενημερωμένο τσιπ ονομάστηκε 68010. Προσθέτει επίσης μια νέα "λειτουργία βρόχου" που επιταχύνει τους μικρούς βρόχους και αυξάνει τη συνολική απόδοση κατά περίπου 10% στις ίδιες ταχύτητες ρολογιού. Μια περαιτέρω διευρυμένη έκδοση, η οποία εκθέτει 31 bits του διαύλου διευθύνσεων, παρήχθη επίσης σε μικρές ποσότητες ως 68012.Για την υποστήριξη συστημάτων χαμηλότερου κόστους και εφαρμογών ελέγχου με μικρότερα μεγέθη μνήμης, η Motorola παρουσίασε τον συμβατό με 8 bit MC68008, επίσης το 1982. Πρόκειται για έναν 68000 με δίαυλο δεδομένων 8 bit και μικρότερο (20 bit) δίαυλο διευθύνσεων. Μετά το 1982, η Motorola αφιέρωσε περισσότερη προσοχή στα έργα 68020 και 88000.

2.5.3 Εφαρμογές

- Το 1981, η Motorola παρουσίασε το Motorola 68000 Educational Computer Board, έναν υπολογιστή μιας πλακέτας για εκπαιδευτικούς και επιμορφωτικούς σκοπούς, ο οποίος εκτός από τον ίδιο τον 68000 περιείχε μνήμη, συσκευές εισόδου/εξόδου,

προγραμματιζόμενο χρονοδιακόπτη και χώρο για προσαρμοσμένα κυκλώματα. Η πλακέτα παρέμεινε σε χρήση σε αμερικανικά κολέγια ως εργαλείο για την εκμάθηση προγραμματισμού συναρμολόγησης μέχρι τις αρχές της δεκαετίας του 1990.

- Ο 68000 γνώρισε επίσης μεγάλη επιτυχία ως ενσωματωμένος ελεγκτής. Ήδη από το 1981, εκτυπωτές λέιζερ όπως ο Imagen Imprint-10 ελέγχονταν από εξωτερικές πλακέτες εξοπλισμένες με τον 68000. Ο πρώτος HP LaserJet, που παρουσιάστηκε το 1984, είχε ενσωματωμένο 68000 8 MHz. Άλλοι κατασκευαστές εκτυπωτών υιοθέτησαν τον 68000, συμπεριλαμβανομένης της Apple με την παρουσίαση του LaserWriter το 1985, του πρώτου εκτυπωτή λέιζερ PostScript. Ο 68000 συνέχισε να χρησιμοποιείται ευρέως σε εκτυπωτές καθ' όλη τη διάρκεια της υπόλοιπης δεκαετίας του 1980, και επέμεινε μέχρι και τη δεκαετία του 1990 σε εκτυπωτές χαμηλού επιπέδου.

- Ο 68000 ήταν επιτυχημένος στον τομέα των βιομηχανικών συστημάτων ελέγχου. Μεταξύ των συστημάτων που επωφελήθηκαν από την ύπαρξη ενός 68000 ή παραγώγου του ως μικροεπεξεργαστή ήταν οικογένειες προγραμματιζόμενων λογικών ελεγκτών (PLC) που κατασκευάστηκαν από την Allen-Bradley, την Texas Instruments και στη συνέχεια, μετά την εξαγορά του εν λόγω τμήματος της TI, από τη Siemens. Οι χρήστες τέτοιων συστημάτων δεν αποδέχονται την απαξίωση των προϊόντων με τον ίδιο ρυθμό όπως οι οικιακοί χρήστες και είναι απολύτως πιθανό ότι, παρά το γεγονός ότι έχουν εγκατασταθεί πριν από 20 χρόνια, πολλοί ελεγκτές με βάση τον 68000 θα συνεχίσουν να λειτουργούν αξιόπιστα και στον 21ο αιώνα.

- Σε ορισμένους ψηφιακούς παλμογράφους από τη δεκαετία του '80, ο 68000 έχει χρησιμοποιηθεί ως επεξεργαστής απεικόνισης κυματομορφών - ορισμένα μοντέλα, συμπεριλαμβανομένου του LeCroy 9400/9400A, χρησιμοποιούν επίσης τον 68000 ως μαθηματικό επεξεργαστή κυματομορφών (συμπεριλαμβανομένης της πρόσθεσης, αφαίρεσης, πολλαπλασιασμού και διαίρεσης δύο κυματομορφών/αναφορών/μνημών κυματομορφών), και ορισμένοι ψηφιακοί παλμογράφοι που χρησιμοποιούν τον 68000 (συμπεριλαμβανομένου του 9400/9400A) μπορούν επίσης να εκτελέσουν λειτουργίες γρήγορου μετασχηματισμού Fourier σε μια κυματομορφή.

2.5.4 Αρχιτεκτονική

Address bus

Ο 68000 διαθέτει έναν εξωτερικό δίαυλο διευθύνσεων 24-bit και δύο σήματα επιλογής byte που "αντικατέστησαν" το A0. Αυτές οι 24 γραμμές μπορούν επομένως να διευθυνσιοδοτήσουν 16 MB φυσικής μνήμης με ανάλυση byte. Η αποθήκευση και ο υπολογισμός διευθύνσεων χρησιμοποιεί εσωτερικά 32 bit- ωστόσο, τα 8 bit διεύθυνσης υψηλής τάξης αγνοούνται λόγω της φυσικής έλλειψης ακροδεκτών της συσκευής. Αυτό επιτρέπει την εκτέλεση λογισμικού γραμμένου για ένα λογικά επίπεδο χώρο διευθύνσεων 32 bit, ενώ έχει πρόσβαση μόνο σε ένα φυσικό χώρο

διευθύνσεων 24 bit. Η πρόθεση της Motorola με τον εσωτερικό χώρο διευθύνσεων των 32 bit ήταν η μελλοντική συμβατότητα, καθιστώντας εφικτή τη συγγραφή λογισμικού 68000 που θα εκμεταλλευόταν πλήρως τις μεταγενέστερες υλοποιήσεις 32 bit του συνόλου εντολών 68000.

Ωστόσο, αυτό δεν εμπόδιζε τους προγραμματιστές να γράψουν προς τα εμπρός ασύμβατο λογισμικό. Το λογισμικό "24-bit" που απέρριπτε το ανώτερο byte διεύθυνσης ή το χρησιμοποιούσε για άλλους σκοπούς εκτός από τη διευθυνσιοδότηση, μπορούσε να αποτύχει σε υλοποιήσεις 32-bit του 68000. Για παράδειγμα, οι πρώιμες εκδόσεις (πριν από την έκδοση 7.0) του Mac OS της Apple χρησιμοποιούσαν το ανώτερο byte των κύριων δεικτών μπλοκ μνήμης για να κρατούν σημαίες όπως locked και purgeable. Οι μεταγενέστερες εκδόσεις του λειτουργικού συστήματος μετέφεραν τις σημαίες σε μια κοντινή θέση και η Apple άρχισε να παραδίδει υπολογιστές που είχαν "καθαρές" ROM 32-bit, αρχής γενομένης από την κυκλοφορία του Mac IIci του 1989. Η οικογένεια 68000 αποθηκεύει ακέραιους αριθμούς πολλών byte στη μνήμη με σειρά big-endian.

Internal registers

Η CPU διαθέτει οκτώ καταχωρητές δεδομένων γενικής χρήσης 32 bit (D0-D7) και οκτώ καταχωρητές διευθύνσεων (A0-A7). Ο τελευταίος καταχωρητής διευθύνσεων είναι ο δείκτης στοίβας και οι assemblers δέχονται την ετικέτα SP ως ισοδύναμη με τον A7. Αυτός ήταν ένας καλός αριθμός καταχωρητών εκείνη την εποχή από πολλές απόψεις. Ήταν αρκετά μικρός ώστε να επιτρέπει στον 68000 να ανταποκρίνεται γρήγορα στις διακοπές (ακόμη και στη χειρότερη περίπτωση όπου και οι 8 καταχωρητές δεδομένων D0-D7 και οι 7 καταχωρητές διευθύνσεων A0-A6 έπρεπε να αποθηκευτούν, δηλαδή 15 καταχωρητές συνολικά), και αρκετά μεγάλος ώστε να κάνει τους περισσότερους υπολογισμούς γρήγορους, επειδή μπορούσαν να γίνουν εξ ολοκλήρου μέσα στον επεξεργαστή χωρίς να διατηρούνται στη μνήμη τυχόν επιμέρους αποτελέσματα. (Σημειώστε ότι μια ρουτίνα εξαιρέσεων σε λειτουργία επόπτη μπορεί επίσης να αποθηκεύσει τον δείκτη στοίβας χρήστη A7, ο οποίος θα ήταν συνολικά 8 καταχωρητές διευθύνσεων. Ωστόσο, η σχεδίαση του 68000 με διπλό δείκτη στοίβας (A7 και A7' σε κατάσταση εποπτείας) καθιστά αυτό συνήθως περιττό, εκτός από την περίπτωση που πραγματοποιείται εναλλαγή εργασιών σε ένα σύστημα πολλαπλών εργασιών).

Status register

Ο 68000 διαθέτει έναν καταχωρητή κατάστασης 16 bit. Τα ανώτερα 8 bit είναι το byte του συστήματος και η τροποποίησή του είναι προνομιακή. Τα κάτω 8 bits είναι το byte χρήστη, επίσης γνωστό ως καταχωρητής κωδικού κατάστασης (CCR), και η τροποποίησή του δεν είναι προνομιακή. Οι συγκριτικές, αριθμητικές και λογικές πράξεις του 68000 τροποποιούν τους κωδικούς κατάστασης για να καταγράψουν τα αποτελέσματά τους για χρήση από μεταγενέστερα άλματα υπό συνθήκη. Τα bits του κωδικού συνθήκης είναι "μηδέν" (Z), "μεταφορά" (C), "υπερχείλιση" (V), "επέκταση" (X) και "αρνητικό" (N). Η σημαία "extend" (X) χρήζει ιδιαίτερης αναφοράς, επειδή είναι ξεχωριστή από τη σημαία carry. Αυτό επιτρέπει το διαχωρισμό του επιπλέον bit από τις αριθμητικές, λογικές και μετατοπιστικές πράξεις από την αριθμητική μεταφοράς πολλαπλής ακρίβειας.

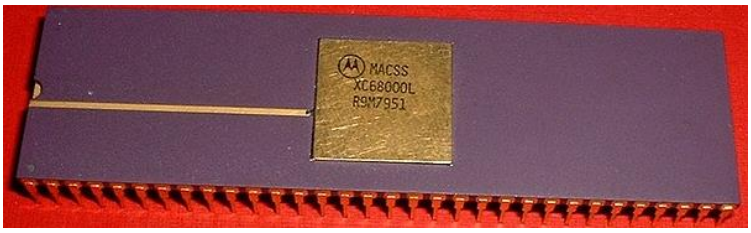
Instruction set

Οι σχεδιαστές προσπάθησαν να καταστήσουν τη γλώσσα συναρμολόγησης ορθογώνια. Δηλαδή, οι εντολές χωρίζονται σε λειτουργίες και τρόπους διεύθυνσης και σχεδόν όλοι οι τρόποι διεύθυνσης είναι διαθέσιμοι για σχεδόν όλες τις εντολές. Υπάρχουν 56 εντολές και ελάχιστο μέγεθος εντολής 16 bit. Πολλές εντολές και τρόποι διευθυνσιοδότησης είναι μακρύτερες για να περιλαμβάνουν περισσότερα bits διεύθυνσης ή τρόπου λειτουργίας.

Interrupts

Η CPU αναγνωρίζει επτά επίπεδα διακοπών. Τα επίπεδα 1 έως 5 έχουν αυστηρή προτεραιότητα. Δηλαδή, μια διακοπή με υψηλότερο αριθμό μπορεί πάντα να διακόψει μια διακοπή με χαμηλότερο αριθμό. Στον καταχωρητή κατάστασης, μια προνομιακή εντολή επιτρέπει τον καθορισμό του τρέχοντος ελάχιστου επιπέδου διακοπής, μπλοκάροντας τις διακοπές χαμηλότερης ή ίσης προτεραιότητας. Για παράδειγμα, εάν το επίπεδο διακοπής στον καταχωρητή κατάστασης έχει οριστεί στο 3, τα υψηλότερα επίπεδα από το 4 έως το 7 μπορούν να προκαλέσουν μια εξαίρεση. Το επίπεδο 7 είναι μια διακοπή επιπέδου που ενεργοποιείται με μη μασκοποιήσιμη διακοπή (NMI). Το επίπεδο 1 μπορεί να διακοπεί από οποιοδήποτε υψηλότερο επίπεδο. Το επίπεδο 0 σημαίνει ότι δεν υπάρχει διακοπή. Το επίπεδο αποθηκεύεται στον καταχωρητή κατάστασης και είναι ορατό στα προγράμματα επιπέδου χρήστη. Οι διακοπές υλικού σηματοδοτούνται στην ΚΜΕ χρησιμοποιώντας τρεις εισόδους που κωδικοποιούν την υψηλότερη εκκρεμή προτεραιότητα διακοπής. Συνήθως απαιτείται ξεχωριστός κωδικοποιητής για την κωδικοποίηση των διακοπών, αν και για συστήματα που δεν απαιτούν περισσότερες από τρεις διακοπές υλικού είναι δυνατόν να συνδεθούν τα σήματα διακοπών απευθείας στις κωδικοποιημένες εισόδους με κόστος μεγαλύτερης πολυπλοκότητας του λογισμικού.

2.5.5 Motorola 68000



Ο μικροεπεξεργαστής Motorola 68000 είναι ο πρώτος μικροεπεξεργαστής 16-bit της Motorola. Ο 68000 δεν είναι συμβατός με την οικογένεια μικροεπεξεργαστών 8-bit της Motorola. Για το σχεδιασμό του συνόλου εντολών η Motorola έδωσε μεγαλύτερη έμφαση στην ισχύ και την απλότητά του παρά στη συμβατότητα. Τα χαρακτηριστικά του Motorola 68000 είναι :

- Είναι ένας μικροεπεξεργαστής 16 bit.
- Διαθέτει δεκαεπτά καταχωρητές δεδομένων και διευθύνσεων των 32 bit, έναν μετρητή προγράμματος των 32 bit (από τον οποίο χρησιμοποιούνται μόνο 24 bit) και έναν καταχωρητή κατάστασης των 16 bit.
- Οι καταχωρητές δεδομένων μπορούν να χρησιμοποιηθούν για τη διαχείριση 8-bit bytes, 16-bit words ή 32-bit long words.
- Υπάρχουν επτά καταχωρητές διευθύνσεων γενικής χρήσης (A0-A6). Αυτοί οι καταχωρητές μπορούν να χειριστούν τελεστές είτε λέξης 16-bit είτε λέξης 32-bit είτε μακράς λέξης 32-bit.
- Ο 68000 υποστηρίζει τη διοχέτευση σε σωλήνες (pipelining).
- Ο 68000 δεν επιχειρεί να φιλοξενήσει διαφορετικές πολυπλοκότητες διαμορφώσεων του συστήματος, όπως π.χ. ακροδέκτες διπλής λειτουργίας στον 8086 και διαφορετικές εκδόσεις στον Ο 68000 παρέχει ξεχωριστούς ακροδέκτες για κάθε γραμμή δεδομένων και γραμμή εξόδου διεύθυνσης. Αυτό είναι εφικτό δεδομένου ότι ο 68000 περιέχεται σε συσκευασία 64 ακίδων και ως εκ τούτου δεν υπάρχει έλλειψη συνδέσεων ακίδων.
- Ο 68000 διαθέτει ενσωματωμένη λογική για το χειρισμό της διαιτησίας πρόσβασης στο δίαυλο σε διαμόρφωση πολλαπλών CPU.
- Ο 68000 μπορεί να έχει άμεση πρόσβαση σε έως και 16 Mbytes μνήμης με τη διεύθυνση των 24-bit Η. Ο χώρος μνήμης μπορεί, να επεκταθεί σε 64 Mbytes με τη χρήση των γραμμών κώδικα λειτουργίας.

2.5.6 Οικογένειες επεξεργαστών

MC68EC000FN10

MC68000ZB8

MC68000IL8

2.5.7 Κατάλογος κατασκευαστών

Κεφάλαιο 3: Λειτουργικά Συστήματα Ενσωματωμένων Συστημάτων

3.1 Εισαγωγή στο MBED OS

Το Mbed OS είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα για πλακέτες Internet of Things (IoT) Cortex-M: χαμηλής κατανάλωσης, περιορισμένης ισχύος και σύνδεσης. Παρέχει ένα επίπεδο αφαίρεσης για τους μικροελεκτές στους οποίους εκτελείται, έτσι ώστε οι προγραμματιστές να μπορούν να γράφουν εφαρμογές C/C++ που εκτελούνται σε οποιαδήποτε πλακέτα με δυνατότητα Mbed. Αποτελείται από τις βασικές βιβλιοθήκες που παρέχουν προγράμματα οδήγησης περιφερειακών μικροελεκτών, δικτύωση, RTOS και περιβάλλον χρόνου εκτέλεσης, εργαλεία κατασκευής και σενάρια δοκιμής και εντοπισμού σφαλμάτων. Αυτές οι συνδέσεις μπορούν να ασφαλιστούν με συμβατές βιβλιοθήκες SSL/TLS, όπως το Mbed TLS ή το wolfSSL, που υποστηρίζει mbed-rtos. Μια βάση δεδομένων στοιχείων παρέχει βιβλιοθήκες προγραμμάτων οδήγησης για στοιχεία και υπηρεσίες που μπορούν να συνδεθούν με τους μικροελεκτές για την κατασκευή ενός τελικού προϊόντος.

3.2 Προφίλ και RTOS (Real Time Operation System)

Το πλήρες προφίλ του Embed OS είναι ένα RTOS, επομένως υποστηρίζει ντετερμινιστική, πολυνηματική εκτέλεση λογισμικού σε πραγματικό χρόνο. Τα πρωτόγονα RTOS είναι πάντα διαθέσιμα, επιτρέποντας στα προγράμματα οδήγησης και τις εφαρμογές να βασίζονται σε νήματα, σηματοφόρους και άλλες δυνατότητες RTOS. Περιλαμβάνει επίσης όλα τα API από προεπιλογή.

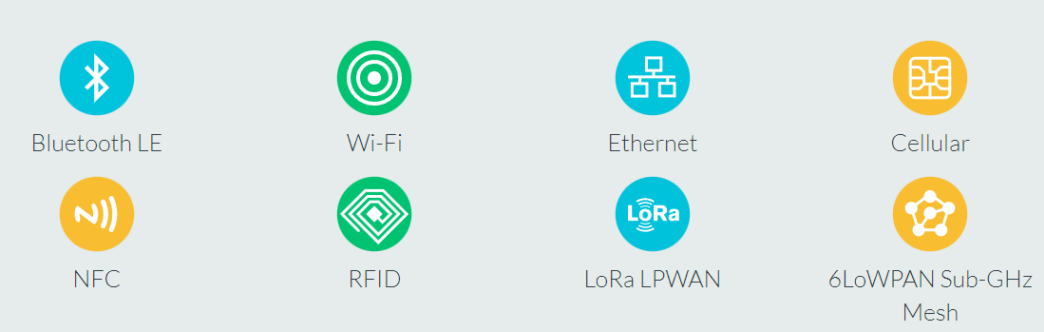
3.3 Χαρακτηριστικά

Modular(Συναρμολογούμενος):Οι απαραίτητες βιβλιοθήκες περιλαμβάνονται αυτόματα στη συσκευή σας, επιτρέποντάς σας να συγκεντρωθείτε στη σύνταξη κώδικα εφαρμογής.

Secure(Ασφαλής):Η ασφάλεια πολλαπλών επιπέδων βοηθά στην προστασία της λύσης IoT σας, από απομονωμένους τομείς ασφαλείας έως το Mbed TLS για ασφαλείς επικοινωνίες.

Connected(Συνδεδεμένος):Σας προσφέρουμε ένα ευρύ φάσμα επιλογών επικοινωνίας με προγράμματα οδήγησης για Bluetooth Low Energy, 6LoWPAN, Mobile IoT (LPWA), Ethernet και WiFi.

3.4 Συνδεσιμότητα



Ένα ευρύ φάσμα επιλογών συνδεσιμότητας είναι διαθέσιμο στο Mbed OS, που υποστηρίζεται από βιβλιοθήκες λογισμικού, υλικό ανάπτυξης, σεμινάρια και παραδείγματα.

3.5 Οφέλη

- Ανοιχτή πηγή(Open Source): Κυκλοφόρησε με άδεια χρήσης Apache 2.0, μπορείτε να χρησιμοποιήσετε το Embed OS σε εμπορικά και προσωπικά έργα με σιγουριά.

- Ευκολία στη χρήση(Ease of Use): Με μια αρθρωτή δομή βιβλιοθήκης, η απαραίτητη υποστήριξη για την εφαρμογή σας θα συμπεριληφθεί αυτόματα στη συσκευή σας. Χρησιμοποιώντας το Embed OS API, ο κώδικας της εφαρμογής σας μπορεί να παραμείνει καθαρός, φορητός και απλός, ενώ εκμεταλλεύεται την ασφάλεια και τις επικοινωνίες. Παρέχουμε μια τεράστια γκάμα παραδειγμάτων κώδικα που μπορείτε να εισάγετε στα δικά σας έργα, δείχνοντας πώς να χρησιμοποιείτε κάθε ένα από τα API.

- Κοινότητα(Community): Η κοινότητα Embed επιτρέπει τη συνεισφορά και τη συνεργασία μεταξύ της Arm, εκατοντάδων συνεργατών και χιλιάδων προγραμματιστών σε όλο τον κόσμο.

- Ασφάλεια από άκρη σε άκρη(End to End Security): Αντιμετωπίζουμε την ασφάλεια στο υλικό της συσκευής, το λογισμικό, την επικοινωνία και τον κύκλο ζωής της ίδιας της συσκευής:

- Επιβεβλημένη ασφάλεια υλικού(Hardware Enforced Security): Ο συνδυασμός συστημάτων πιστοποιημένων PSA, απομόνωσης με επιβολή υλικού με την τεχνολογία Trust Zone και το λογισμικό αναφοράς από το έργο Trusted Firmware-M σάς επιτρέπουν να αναπτύξετε ασφαλείς συσκευές με Embed OS.

- Ασφάλεια Επικοινωνιών(Communications Security): Λαμβάνουμε τα SSL και TLS, τα τυπικά πρωτόκολλα για την ασφάλεια των επικοινωνιών στο Διαδίκτυο, και σας δίνουμε τη δυνατότητα να τα συμπεριλάβετε στο έργο σας Embed με ένα απλό API.

3.6 Πόσο σημαντικό είναι

Η ARM χρησιμοποιεί το όνομα mbed από το 2005 για πλατφόρμες ανάπτυξης τύπου "maker" που βασίζονται σε MCU της σειράς Cortex-M, μαζί με μια μεγάλη κοινότητα προγραμματιστών και μια εκτενή βιβλιοθήκη λογισμικού. Αλλά η νέα ανακοίνωση επεκτείνει σημαντικά την αρχική ιδέα mbed. Το όνομα mbed περιλαμβάνει πλέον όχι μόνο το νέο λειτουργικό σύστημα, αλλά και μια πλατφόρμα συνδεσιμότητας cloud (mbed Device Server) ,ένα σύνολο εργαλείων ανάπτυξης (mbed Tools) και ένα οικοσύστημα εταίρων (mbed Partners). Ουσιαστικά, το mbed έχει γίνει μια σειρά προϊόντων και υπηρεσιών. Η ARM λέει ότι συλλογικά, το mbed θα «επιταχύνει την ανάπτυξη του Internet of Things».

1. Πρώτον, το δωρεάν λειτουργικό σύστημα θα είναι εμπορικής ποιότητας. Προσφέροντάς το δωρεάν, το mbed OS θα ανταγωνιστεί ορισμένα από τα εμπορικά ενσωματωμένα λειτουργικά συστήματα που κυκλοφορούν ήδη στην αγορά. Ωστόσο, το λειτουργικό σύστημα mbed δεν θα είναι ένα λειτουργικό σύστημα σε πραγματικό χρόνο (RTOS). Πολλές συσκευές IoT απαιτούν τον κρίσιμο χρονικά ντετερμινισμό ενός RTOS, κυρίως σε κρίσιμες για την ασφάλεια εφαρμογές όπως τα ηλεκτρονικά συστήματα, τα συστήματα αυτοκινήτου, ο αυτοματισμός εργοστασίων και τα παρόμοια. Η έλλειψη λειτουργιών σε πραγματικό χρόνο θα περιορίσει το εύρος της εφαρμογής για το mbed OS.

2. Δεύτερον, η ARM είπε ότι η κύρια πρόθεσή της να κυκλοφορήσει το λειτουργικό σύστημα μαζί με τον mbed Device Server ήταν να διευκολύνει την ανάπτυξη ενσωματωμένου λογισμικού για να χειριστεί τις πολλές ανησυχίες ασφαλείας και τα πρωτόκολλα επικοινωνίας που χρησιμοποιούνται στο IoT, καθώς αυτά είναι συχνά εμπόδια για προγραμματιστές που δεν είχαν προηγουμένως εμπειρία με συνδεδεμένες συσκευές .

3. Τρίτον, αν και η ARM δεν το ανέφερε αυτό στις πληροφορίες τύπου της, η Shelby είπε στην VDC ότι μεγάλο μέρος του πηγαίου κώδικα του mbed OS θα διατίθεται ως ανοιχτού κώδικα.

3.7 Η διαφορά του με άλλα συστήματα

- Είναι μια πλατφόρμα ανάπτυξης ταχείας εκκίνησης που περιλαμβάνει συμβατότητα λογισμικού στο IoT ως «πλατφόρμα συσκευών IoT». Επιπλέον, περιλαμβάνει έναν αριθμό διαθέσιμων βιβλιοθηκών, συμπεριλαμβανομένης της τρέχουσας εστίασης στην παροχή δυνατότητας σύνδεσης συσκευών IoT στο cloud.

- Είναι ελκυστικό επειδή πολλά από τα στοιχεία του είναι ανοιχτού κώδικα ή διαθέσιμα δωρεάν. Επίσης, όπως υπονοεί η λέξη "πλατφόρμα", το Mbed δεν είναι μόνο μια πλακέτα μικροελεκτή, εργαλεία ανάπτυξης, λογισμικό και συμβατότητα, αλλά και ένα περιβάλλον ανάπτυξης ή βάση που περιλαμβάνει όλα αυτά. Επιπλέον, οι περισσότερες από τις πλακέτες μικροελεκτών που αναπτύχθηκαν από την Mbed είναι Arm Cortex-M, επομένως υπάρχει πολύ υψηλός βαθμός συμβατότητας.

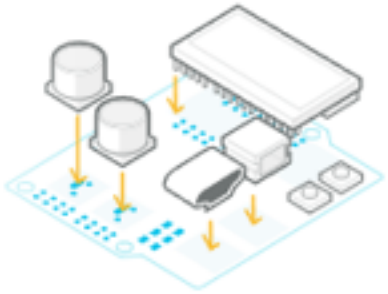
3.8 ARM CORTEX-M

Το ARM Cortex-M είναι μια ομάδα πυρήνων επεξεργαστών RISC ARM 32-bit με άδεια χρήσης από την Arm Holdings. Αυτοί οι πυρήνες είναι βελτιστοποιημένοι για χαμηλού κόστους και ενεργειακά αποδοτικά ολοκληρωμένα κυκλώματα, τα οποία έχουν ενσωματωθεί σε δεκάδες δισεκατομμύρια καταναλωτικές συσκευές. Αν και είναι συνήθως το κύριο συστατικό των τσιπ μικροελεκτών, μερικές φορές ενσωματώνονται και σε άλλους τύπους τσιπ. Η οικογένεια Cortex-M αποτελείται από Cortex-M0, Cortex-M0+, Cortex-M1, Cortex-M3, Cortex-M4, Cortex-M7, Cortex-M23, Cortex-M33, Cortex-M35P, Cortex-M55. Οι πυρήνες Cortex-M4 / M7 / M33 / M35P / M55 διαθέτουν επιλογή πυριτίου FPU και όταν περιλαμβάνονται στο πυρίτιο, αυτοί οι πυρήνες είναι μερικές φορές γνωστοί ως "Cortex-Mx with FPU" ή "Cortex-MxF", όπου το "x" είναι η βασική παραλλαγή.

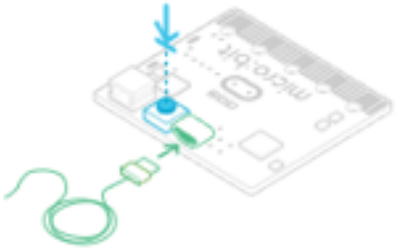
3.9 Κατηγορίες προγραμμάτων

BOARDS(ΠΛΑΚΕΤΕΣ): Εστιάζοντας στην αξιολόγηση και την ταχεία δημιουργία πρωτοτύπων, οι πίνακες είναι πλατφόρμες ανάπτυξης που ενσωματώνουν μικροελεγκτές ή μονάδες με εξαρτήματα για να δημιουργήσουν ένα σύστημα στο οποίο λειτουργεί το Mbed OS. Τις περισσότερες φορές συνοδεύεται από κύκλωμα εντοπισμού σφαλμάτων και επιλογές περιφερειακής σύνδεσης.

COMPONETS(ΕΞΑΡΤΗΜΑΤΑ): Τα στοιχεία υλικού επεκτείνουν τις δυνατότητες μικροελεγκτών, μονάδων ή πλακών και περιλαμβάνουν τις σχετικές βιβλιοθήκες λογισμικού για το Mbed OS.



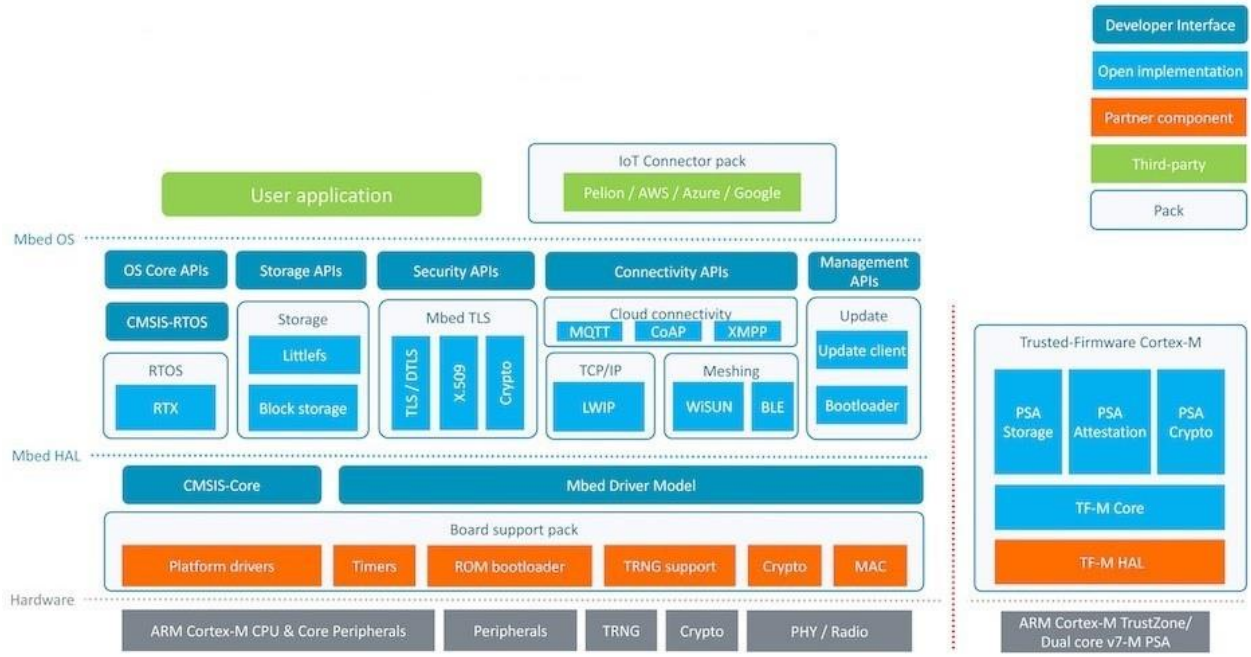
INTERFACES(ΔΙΕΠΑΦΕΣ): Κυκλώματα υλικού και συνοδευτικά υλικολογισμικά που χρησιμοποιούνται για την ανάπτυξη, τον εντοπισμό σφαλμάτων και τη δοκιμή λογισμικού που εκτελείται σε μικροελεγκτές βασισμένους σε βραχίονα.



3.10 RTOS (Real Time Operation System)

Ένα λειτουργικό σύστημα σε πραγματικό χρόνο (RTOS) είναι ένα λειτουργικό σύστημα (OS) για εφαρμογές σε πραγματικό χρόνο που επεξεργάζεται δεδομένα και συμβάντα που έχουν κρίσιμους χρονικούς περιορισμούς. Ένα RTOS διαφέρει από ένα λειτουργικό σύστημα κοινής χρήσης χρόνου, όπως το Unix, το οποίο διαχειρίζεται την κοινή χρήση των πόρων του συστήματος με έναν χρονοπρογραμματιστή, buffer δεδομένων ή καθορισμένη ιεράρχηση εργασιών σε περιβάλλον πολλαπλών εργασιών ή πολυπρογραμματισμού. Οι απαιτήσεις χρόνου επεξεργασίας πρέπει να είναι πλήρως κατανοητές και να δεσμεύονται και όχι απλώς να διατηρούνται στο ελάχιστο. Τα λειτουργικά συστήματα σε πραγματικό χρόνο βασίζονται σε συμβάντα και προληπτικά, πράγμα που σημαίνει ότι το ΛΣ είναι ικανό να παρακολουθεί τη σχετική προτεραιότητα των ανταγωνιστικών εργασιών και να κάνει αλλαγές στην προτεραιότητα εργασίας. Τα συστήματα που βασίζονται σε συμβάντα αλλάζουν μεταξύ εργασιών με βάση τις προτεραιότητές τους, ενώ τα συστήματα χρονομερισμού αλλάζουν την εργασία με βάση τις διακοπές ρολογιού.

3.11 Αρχιτεκτονική Embed OS



3.12 Τα θεμέλια

Το Mbed OS χρησιμοποιεί ένα επίπεδο αφαίρεσης υλικού (HAL) για την υποστήριξη των πιο κοινών τμημάτων ενός μικροελεκτή, όπως οι χρονοδιακόπτες. Αυτό το θεμέλιο διευκολύνει τη συγγραφή εφαρμογών με βάση ένα κοινό σύνολο διεπαφών προγραμματισμού εφαρμογών (APIs)- η συσκευή σας περιλαμβάνει αυτόματα τις απαραίτητες βιβλιοθήκες και την υποστήριξη προγραμμάτων οδήγησης για τυποποιημένες περιφερειακές μονάδες MCU, όπως I2C, Bus και SPI.

Το HAL χρησιμεύει επίσης ως σημείο εκκίνησης κατά την προσθήκη υποστήριξης για νέους στόχους ή χαρακτηριστικά σε υπάρχοντες στόχους. Συνεργαζόμαστε στενά με τους συνεργάτες μας σε πυρίτιο για τη μεταφορά αυτών των νέων χαρακτηριστικών σε αναπτυξιακές πλακέτες με δυνατότητα Mbed.

Η δομή του Mbed OS επιτρέπει την αντιστοίχιση εφαρμογών και συστημάτων αποθήκευσης. Με άλλα λόγια, όπου οι επιλογές αποθήκευσης σε επίπεδο μπλοκ διαφέρουν και εξαρτώνται από την εφαρμογή, μπορείτε να επιλέξετε το σύστημα αρχείων που ταιριάζει καλύτερα στη συσκευή IoT σας. Το σύστημα αρχείων FAT - που υποστηρίζεται από μια κάρτα SD - παρέχει συμβατότητα με άλλα λειτουργικά συστήματα, όπως τα Windows, το Mac OS ή το Linux. Όταν είναι σημαντική η υψηλή αξιοπιστία και η αποκατάσταση από διακοπή ρεύματος, έχει νόημα να χρησιμοποιήσετε το ενσωματωμένο σύστημα αρχείων μας, το οποίο υποστηρίζεται από ένα τσιπ (Q)SPI NOR flash. Τέλος, το Mbed OS υλοποιεί για εσάς το επίπεδο επαναπροσδιορισμού και την ενσωμάτωση της

διαδικασίας εκκίνησης κάθε υποστηριζόμενης αλυσίδας εργαλείων, ώστε η ανάπτυξη εφαρμογών να μοιάζει με την ανάπτυξη σε C ή C++ για οποιοδήποτε άλλο λειτουργικό σύστημα.

3.2.1 Εισαγωγή στο FreeRTOS

Το FreeRTOS, που αναπτύχθηκε σε συνεργασία με τις κορυφαίες εταιρείες τσιπ παγκοσμίως επί 18 χρόνια, είναι ένα κορυφαίο λειτουργικό σύστημα πραγματικού χρόνου (RTOS) για μικροελεγκτές και μικρούς μικροεπεξεργαστές. Διανέμεται ελεύθερα υπό την άδεια ανοικτού κώδικα MIT, το FreeRTOS περιλαμβάνει έναν πυρήνα και ένα αυξανόμενο σύνολο βιβλιοθηκών IoT κατάλληλων για χρήση σε όλους τους βιομηχανικούς τομείς. Το FreeRTOS έχει κατασκευαστεί με έμφαση στην αξιοπιστία και την ευκολία χρήσης.

3.2.2 Γιατί χρησιμοποιούμε το FreeRTOS

- Αξιόπιστος πυρήνας

Με αποδεδειγμένη ανθεκτικότητα, μικρό αποτύπωμα και ευρεία υποστήριξη συσκευών, ο πυρήνας FreeRTOS εμπιστεύονται κορυφαίες εταιρείες παγκοσμίως ως το de facto πρότυπο για μικροελεγκτές και μικρούς μικροεπεξεργαστές.

- Επιτάχυνση του χρόνου διάθεσης στην αγορά

Με λεπτομερείς προ-ρυθμισμένες επιδείξεις και ενσωματώσεις αναφοράς για το Διαδίκτυο των πραγμάτων (IoT), δεν χρειάζεται να καθορίσετε τον τρόπο ρύθμισης ενός έργου. Πραγματοποιήστε γρήγορη λήψη, μεταγλώττιση και βγείτε ταχύτερα στην αγορά.

- Ευρεία υποστήριξη του οικοσυστήματος

Το οικοσύστημα συνεργατών μας παρέχει ένα ευρύ φάσμα επιλογών, συμπεριλαμβανομένων των συνεισφορών της κοινότητας, της επαγγελματικής υποστήριξης, καθώς και των ολοκληρωμένων εργαλείων IDE και παραγωγικότητας.

- Προβλεψιμότητα της μακροπρόθεσμης στήριξης

Το FreeRTOS προσφέρει σταθερότητα χαρακτηριστικών με εκδόσεις μακροχρόνιας υποστήριξης (LTS). Οι βιβλιοθήκες FreeRTOS LTS(Long Term Support) συνοδεύονται από ενημερώσεις ασφαλείας και κρίσιμες διορθώσεις σφαλμάτων για δύο χρόνια. Συντηρείται από την AWS προς όφελος της κοινότητας του FreeRTOS.

3.2.3 Χαρακτηριστικά

- Μικροσκοπικός πυρήνας εξοικονόμησης ενέργειας : Κλιμακούμενο μέγεθος, με ωφέλιμο αποτύπωμα μνήμης προγράμματος μόλις 9KB. Ορισμένες αρχιτεκτονικές περιλαμβάνουν μια λειτουργία εξοικονόμησης ενέργειας

- Υποστήριξη για 40+ αρχιτεκτονικές : Μία βάση κώδικα για 40+ αρχιτεκτονικές MCU και 15+ αλυσίδες εργαλείων, συμπεριλαμβανομένων των τελευταίων μικροελεκτών RISC-V και ARMv8-M (Arm Cortex-M33).

- Αρθρωτές βιβλιοθήκες : Ένας αυξανόμενος αριθμός πρόσθετων βιβλιοθηκών που χρησιμοποιούνται σε όλους τους βιομηχανικούς τομείς, συμπεριλαμβανομένης της ασφαλούς τοπικής συνδεσιμότητας ή συνδεσιμότητας στο cloud

- Ενσωματώσεις αναφοράς AWS : Επωφελούμαστε από δοκιμασμένα παραδείγματα που περιλαμβάνουν όλες τις βιβλιοθήκες που είναι απαραίτητες για την ασφαλή σύνδεση στο cloud

- Με άδεια MIT, με επιλογές : Το FreeRTOS μπορεί να χρησιμοποιηθεί για οποιονδήποτε σκοπό σύμφωνα με την άδεια χρήσης MIT. Ο στρατηγικός μας συνεργάτης παρέχει επίσης εμπορικές άδειες και πιστοποίηση ασφάλειας.

3.3.1 Εισαγωγή στο QNX

Αναπτύχθηκε αρχικά από την Quantum Software Systems στις αρχές της δεκαετίας του '80. Τώρα ανήκει στην Blackberry. Το QNX είναι ένα ενσωματωμένο RTOS που προορίζεται για την ανάπτυξη συστημάτων κρίσιμων αποστολών και είναι ένα λειτουργικό σύστημα βασισμένο σε μικροπυρήνα που μοιάζει με το UNIX. Μπορούμε να δημιουργήσουμε ένα ενσωματωμένο σύστημα που διασφαλίζει ότι η εφαρμογή μας ολοκληρώνει τις εργασίες της εντός του χρονικού πλαισίου που απαιτεί το έργο μας. Το σύστημα χρησιμοποιείται κυρίως στους τομείς της βιομηχανίας, της αυτοκινητοβιομηχανίας και της ιατρικής.

3.3.2 Πλεονεκτήματα και Μειονεκτήματα

Πλεονεκτήματα του ενσωματωμένου συστήματος QNX

- Είναι ένα λειτουργικό σύστημα πραγματικού χρόνου
- Υποστηρίζει πολλές αρχιτεκτονικές
- Μπορείτε να μεταφέρετε εφαρμογή Linux στο QNX
- Υποστηρίζει Qt5 και Felgo

Μειονεκτήματα του ενσωματωμένου συστήματος QNX

- Έχει ιδιωτική άδεια χρήσης
- Είναι αρκετά ακριβό

3.3.3 Ασφάλεια, επιδόσεις και χαρακτηριστικά

Το QNX είναι ένα ενσωματωμένο λειτουργικό σύστημα πραγματικού χρόνου που εστιάζει στην απόδοση και την αξιοπιστία. Τα χαρακτηριστικά του είναι σχεδιασμένα ώστε η ενσωματωμένη συσκευή να εκτελεί το έργο της χωρίς σφάλματα και εγκαίρως.

- Ορισμένα από τα πιο σημαντικά χαρακτηριστικά του QNX είναι τα εξής:
- Αρχιτεκτονική του μικροπυρήνα
- Υποστήριξη πολυπύρηνων
- Διαχειριστής σύνθεσης βελτιστοποιημένου υλικού που υποστηρίζει HTML5, Kanzi, OpenGL ES, Qt 5, Storyboard, Video και άλλες τεχνολογίες HMI 3ου μέρους
- Υποστηρίζει είσοδο πολλαπλής αφής και λήψη βίντεο
- Ασφαλής εκκίνηση
- Κρυπτογραφημένο σύστημα αρχείων
- Αυτοεπαληθευόμενο σύστημα αρχείων
- Προστασία σωρού, στοίβας και μνήμης

3.3.4 Εργαλεία

Για το QNX μπορείτε να επιλέξετε μεταξύ C++, HTML5, Qt, Python ή Java ως γλώσσα προγραμματισμού για την ανάπτυξη της εφαρμογής σας. Επίσης, μπορείτε να μεταφέρετε τις εφαρμογές σας Linux στο QNX. Μπορείτε να επιλέξετε το QNX SDP και το QNX Momentics Tool Suite με το Eclipse ή το Qt Creator ως εργαλείο ανάπτυξης.

3.3.5 Παραδείγματα πραγματικής ζωής του συστήματος

Το QNX χρησιμοποιείται στους ακόλουθους τομείς:

- Αυτοκίνητο
- Βαρέα μηχανήματα
- Βιομηχανικός έλεγχος
- Ιατρικές συσκευές
- Σιδηροδρομικές γραμμές
- Ρομποτική

3.4.1 Εισαγωγή στο VxWorks

Πρόκειται για ένα RTOS που αναπτύχθηκε από την Wind River Systems. Το VxWorks είναι ένα ιδιόκτητο σύστημα που έχει σχεδιαστεί για διάφορες βιομηχανίες όπως η αεροδιαστημική και η άμυνα, οι ιατρικές συσκευές, ο βιομηχανικός εξοπλισμός, η ρομποτική, η ενέργεια, οι μεταφορές, οι υποδομές δικτύων, η αυτοκινητοβιομηχανία και τα καταναλωτικά ηλεκτρονικά.

3.4.2 Ασφάλεια, επιδόσεις και χαρακτηριστικά

Παρόλο που το VxWorks διαθέτει πολλά χαρακτηριστικά ασφαλείας, πέρυσι το ARMIS αποκάλυψε 11 κρίσιμα τρωτά σημεία του συστήματος, όπως απομακρυσμένη εκτέλεση κώδικα, άρνηση παροχής υπηρεσιών, διαρροή πληροφοριών και λογικά σφάλματα.

Τα VxWorks έχουν τα ακόλουθα χαρακτηριστικά:

- Υποστήριξη μονοπύρηνων και πολυπύρηνων
- Ασφαλής εκκίνηση, φορτωτής ELF και αποθήκευση
- Σκλήρυνση πυρήνα
- Συμβάντα ασφαλείας
- Ενσωματωμένοι έλεγχοι πρόσβασης
- Προηγμένη διαχείριση χρηστών
- Κρυπτογραφία
- Πρωτόκολλα ασφαλείας δικτύου

3.4.3 Εργαλεία

Για την ανάπτυξη εφαρμογών VxWorks υπάρχουν δύο εργαλεία της Wind River Systems:

Το πρώτο είναι το Wind River Workbench, ένα IDE βασισμένο στο Eclipse. Το δεύτερο είναι το Wind River Simics, ένας προσομοιωτής που θα μας επιτρέψει να εικονικοποιήσουμε το υλικό μας και να διευκολύνουμε τη διαδικασία ανάπτυξης. Διαθέτει επίσης εργαλεία συνεχούς ολοκλήρωσης.

Τα εργαλεία ανάπτυξης του VxWorks μας επιτρέπουν να χρησιμοποιήσουμε τα πρότυπα C++ 11 και C++ 14, καθώς και βιβλιοθήκες πολλαπλών πλατφορμών όπως η Boost, το OpenCV και το Qt5 για τη δημιουργία εφαρμογών.

3.4.4 Πραγματικά παραδείγματα του συστήματος

Το VxWorks χρησιμοποιείται σε έναν μακρύ κατάλογο ενσωματωμένων συσκευών. Ένα παράδειγμα είναι το ρομπότ Asimo της Honda ή το Curiosity Rovers της NASA. Αυτοί είναι μερικοί ακόμη τομείς όπου το ενσωματωμένο σύστημα έχει χρησιμοποιηθεί μέχρι σήμερα:

- Διαστημόπλοια: Curiosity Rover, το Mars Reconnaissance Orbiter, ο διαστημικός ανιχνευτής Deep Impact, το SpaceX Dragon.
- Αεροσκάφη: Airbus A400M Airlifter, Boeing 787
- Διαστημικά τηλεσκόπια
- Βιομηχανικά ρομπότ
- Δοκιμές και μετρήσεις
- Μεταφορές
- Ελεγκτές
- Συστήματα αποθήκευσης
- Απεικόνιση
- Ιατρική

3.4.5 Πλεονεκτήματα και Μειονεκτήματα

Πλεονεκτήματα του λειτουργικού συστήματος VxWorks

- Διαθέτει προσομοιωτή
- Υποστηρίζει βιβλιοθήκες πολλαπλών πλατφορμών
- Παρέχει υποστήριξη για πολλές αρχιτεκτονικές 32 και 64 bit.
- Δοκιμάζει σε πολλές συσκευές
- Η τεκμηρίωση είναι διαθέσιμη στον ιστότοπο τους

Μειονεκτήματα του λειτουργικού συστήματος VxWorks

- Διατίθενται μόνο λίγες γλώσσες προγραμματισμού
- Έχει ένα μικρό πρόβλημα ασφάλειας

3.5.1 Εισαγωγή στο Linux Embedded

Το Embedded Linux είναι ένα λειτουργικό σύστημα ανοικτού κώδικα που έχει σχεδιαστεί ειδικά για την έξυπνη άκρη. Το ενσωματωμένο Linux βασίζεται στον ίδιο πυρήνα Linux, διαθέσιμο από την ιστοσελίδα kernel.org, όπως όλα τα συστήματα Linux. Αλλά τα ενσωματωμένα συστήματα έχουν αυστηρούς περιορισμούς που τα συστήματα επιχειρήσεων απλά δεν έχουν, που κυμαίνονται από υψηλότερες απαιτήσεις αξιοπιστίας και ασφάλειας μέχρι αυστηρότερη διαθεσιμότητα πόρων και την ανάγκη για μηχανική υποστήριξη που συχνά διαρκεί 10 χρόνια ή περισσότερο. Επιπλέον, οι ενσωματωμένες συσκευές πρέπει συνήθως να επικοινωνούν μεταξύ τους και με τους πόρους του cloud.

Το Linux για ενσωματωμένα συστήματα, λοιπόν, απαιτεί πρόσθετα πακέτα πέραν του αρχικού πυρήνα. Το ποια συγκεκριμένα πακέτα απαιτούνται για τη διανομή σας εξαρτάται από το τι πρόκειται να κατασκευάσετε και η καλύτερη λύση θα διαφέρει για διαφορετικές περιπτώσεις

χρήσης. Τελικά, κάθε ενσωματωμένο σύστημα Linux είναι μοναδικό, κατασκευασμένο ειδικά για την έξυπνη άκρη.

3.5.2 Γιατί χρησιμοποιούμε το embedded Linux

Το επόμενο ερώτημα που προκύπτει είναι, τι κάνει το Linux τον τέλειο υποψήφιο για ένα ενσωματωμένο σύστημα; Από την επεκτασιμότητα μέχρι την υποστήριξη προγραμματιστών και την ανάπτυξη εργαλείων, υπάρχουν αμέτρητοι λόγοι για τους οποίους αυτό ισχύει. Μια σε βάθος συζήτηση για τον καθένα απαιτεί το δικό του βιβλίο. Εδώ, θα περιοριστούμε να αναφέρουμε μερικούς εξέχοντες από αυτούς:

- Υποστήριξη υλικού
- Δικτύωση
- Αρθρωτότητα
- Εμπορική υποστήριξη

3.5.3 Πλεονεκτήματα και Μειονεκτήματα

Πλεονεκτήματα του λειτουργικού συστήματος embedded Linux

- Διασταυρούμενη μεταγλώττιση για οποιαδήποτε υποστηριζόμενη πλατφόρμα
- Κοινοτική αντανάκλαση των διορθώσεων κοινών τρωτών σημείων και ανοιγμάτων (CVE) σε ενημερωμένες εκδόσεις
- Ανάπτυξη σε ευρέως χρησιμοποιούμενες υποδομές και εργαλεία Linux
- Σύγχρονο, cloud-native περιβάλλον
- Ευρεία υποστήριξη υλικού
- Κύκλος ζωής προϊόντος μέσω κοινοτικών LTS

Μειονεκτήματα του λειτουργικού συστήματος embedded Linux

- Μπορεί να είναι δύσκολο να χρησιμοποιηθεί
- Πρέπει να εφαρμόσετε την ασφάλεια
- Εάν επιλέξετε να φτιάξετε το δικό σας Linux, η απόδοση θα εξαρτηθεί από τον τρόπο με τον οποίο θα ρυθμίσετε το σύστημά σας

3.5.4 Ενσωματωμένη ανάπτυξη Linux

Ανάπτυξη BSP

Ένα πακέτο υποστήριξης πλακέτας (BSP) είναι ένα επίπεδο λογισμικού που περιέχει οδηγούς και άλλες ρουτίνες που αφορούν το υλικό και επιτρέπουν στο Linux να λειτουργεί σε ένα συγκεκριμένο περιβάλλον υλικού. Θα πρέπει να προσαρμόζεται όχι μόνο στο σωστό συνδυασμό αρχιτεκτονικής/OS, αλλά και στην ακριβή διαμόρφωση του συστήματος-σε-τσιπ (SoC) και της πλακέτας. Για τη βελτιστοποίηση ενός ενσωματωμένου περιβάλλοντος, το BSP θα πρέπει επίσης να περιλαμβάνει μια προσαρμοσμένη έκδοση του Linux (ή άλλου λειτουργικού συστήματος). Ο προμηθευτής BSP σας μπορεί να παρέχει προσαρμοσμένες σειρές δοκιμών, βοήθεια προγραμματιστών από εμπειρογνώμονες και άλλες επαγγελματικές υπηρεσίες.

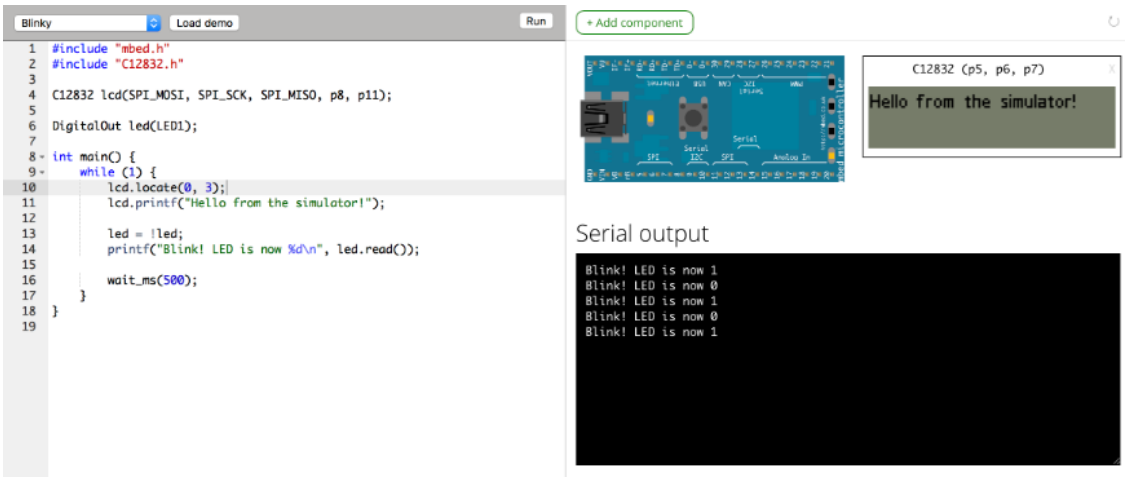
Ολοκλήρωση συστήματος

Μια συνάθροιση υποσυστημάτων συνεργάζεται έτσι ώστε το σύστημα να μπορεί να παρέχει συνολική λειτουργικότητα και να διασφαλίζει ότι τα υποσυστήματα λειτουργούν μαζί ως σύστημα. Η ολοκλήρωση συστήματος περιλαμβάνει το συντονισμό των υφιστάμενων, συχνά διαφορετικών, ενσωματωμένων συστημάτων, έτσι ώστε να μπορεί να αντιμετωπιστεί η εξειδικευμένη περίπτωση χρήσης.

Κεφάλαιο 4: Embed simulator

4.1 Εισαγωγή

Το Embed simulator είναι μια πλατφόρμα και ένα λειτουργικό σύστημα για συσκευές συνδεδεμένες στο διαδίκτυο που βασίζονται σε μικροελεγκτές ARM Cortex-M 32-bit. Τέτοιες συσκευές είναι επίσης γνωστές ως συσκευές Internet of Things. Το έργο αναπτύσσεται σε συνεργασία από την Arm και τους τεχνολογικούς εταίρους της. Ο προσομοιωτής σας επιτρέπει να εκτελείτε τις εφαρμογές σας Mbed OS 5 απευθείας στον υπολογιστή σας, ώστε να μπορείτε να δοκιμάζετε και να επαληθεύετε γρήγορα τις εφαρμογές χωρίς να τις αναβοσβήνετε σε μια πραγματική πλακέτα. Αυτό είναι ένα πολύτιμο εργαλείο εκμάθησης, καθώς μαθαίνετε γρήγορα πώς λειτουργεί το Mbed. Είναι επίσης πολύ χρήσιμο για την ανάπτυξη σύνθετων εφαρμογών. Κυκλοφορεί σε δύο εκδοχές, μια online έκδοση, η οποία εκτελείται εξ ολοκλήρου στο πρόγραμμα περιήγησής σας, και μια offline έκδοση, η οποία λειτουργεί για οποιοδήποτε έργο Mbed OS 5.



Εικόνα 3: Embed Simulator Interface

Λειτουργία

Σε αντίθεση με τα γρήγορα μοντέλα, ο προσομοιωτής Mbed δεν προσομοιώνει πλήρως μια συσκευή Cortex-M. Αντ' αυτού, μεταγλωττίζει το Mbed OS 5 χρησιμοποιώντας το Emscripten. Το Emscripten είναι ένας μεταγλωττιστής LLVM σε JavaScript, ο οποίος παίρνει ένα σύνθετο έργο C++ και παράγει κάτι που μπορεί να εκτελεστεί σε περιβάλλον ιστού. Στη συνέχεια, όταν πρέπει να συμβεί κάτι με τη φυσική πλακέτα (για παράδειγμα, να τραβήξει ένα pin high), η κλήση αναχαιτίζεται και αντί να αλλάζουμε έναν καταχωρητή, μπορούμε να αλλάξουμε ένα γραφικό στοιχείο στην οθόνη. Αυτό γίνεται με την προσθήκη ενός νέου στόχου στο Mbed OS. Μια παρόμοια προσέγγιση ακολουθείται για τα στοιχεία. Τη στιγμή που πρέπει να πραγματοποιηθεί η πραγματική επικοινωνία, αναχαιτίζουμε την κλήση και στέλνουμε τα δεδομένα στο πρόγραμμα περιήγησης. Για παράδειγμα, με μια οθόνη LCD, αναχαιτίζουμε τη στιγμή που ο ρυθμιστής πλαισίων ξεπλένεται και αντ' αυτού απορρίπτουμε την εικόνα σε έναν καμβά. Όλα τα υπόλοιπα στο πρόγραμμα οδήγησης, από την απόδοση γραμματοσειρών μέχρι το API, παραμένουν ακριβώς τα ίδια.

Περιορισμοί

1. Το busy-looping (while (1) {}) χωρίς να καλέσετε wait οπουδήποτε θα κάνει το πρόγραμμα περιήγησης να κολλήσει.

2. Βεβαιωθείτε ότι χρησιμοποιείτε την wait() όταν θέλετε να κάνετε παύση.

3. Δεν είναι καθόλου ακριβής ως προς τον κύκλο. Η εφαρμογή εκτελείται όσο το δυνατόν γρηγορότερα.

4. Πολύ περισσότερη διαθέσιμη μνήμη. Δεν υπάρχουν περιορισμοί για το μέγεθος της στοίβας και του σωρού.

5. Οι διακοπές είναι ψεύτικες. Το πρόγραμμα περιήγησης δεν έχει καμία έννοια των διακοπών, οπότε οι διακοπές δεν έχουν προτεραιότητα έναντι του κύριου νήματος.

6. Δεν υπάρχει RTOS. Πρόκειται για ένα περιβάλλον με ένα μόνο νήμα. Μπορείτε να χρησιμοποιήσετε το mbed-events για να διευκολύνετε λίγο την αντιμετώπιση πολύπλοκων περιβαλλόντων, αλλά δεν μπορείτε να δημιουργήσετε πολλαπλά νήματα.

4.2 Παραδείγματα με κώδικα

HTTP: Hypertext Transfer Protocol (Πρωτόκολλο μεταφοράς υπερκειμένου)

Πρωτόκολλο στο οποίο ανταλλάσσονται πόροι μεταξύ συσκευών-πελατών και διακομιστών μέσω του διαδικτύου. Οι συσκευές-πελάτες στέλνουν αιτήματα σε διακομιστές για τους πόρους που απαιτούνται για τη φόρτιση μιας ιστοσελίδας. Οι διακομιστές στέλνουν απαντήσεις πίσω στον πελάτη για την εκπλήρωση των αιτημάτων.

Μέθοδοι GET και POST

-Αίτημα GET: Αναμένει πληροφορίες σε αντάλλαγμα.

-Αίτημα POST: Υποδεικνύει ότι ο πελάτης, υποβάλλει πληροφορίες στον διακομιστή ιστού π.χ. πληροφορίες φόρμας.

HTTP request headers (αίτηση επικεφαλίδας)

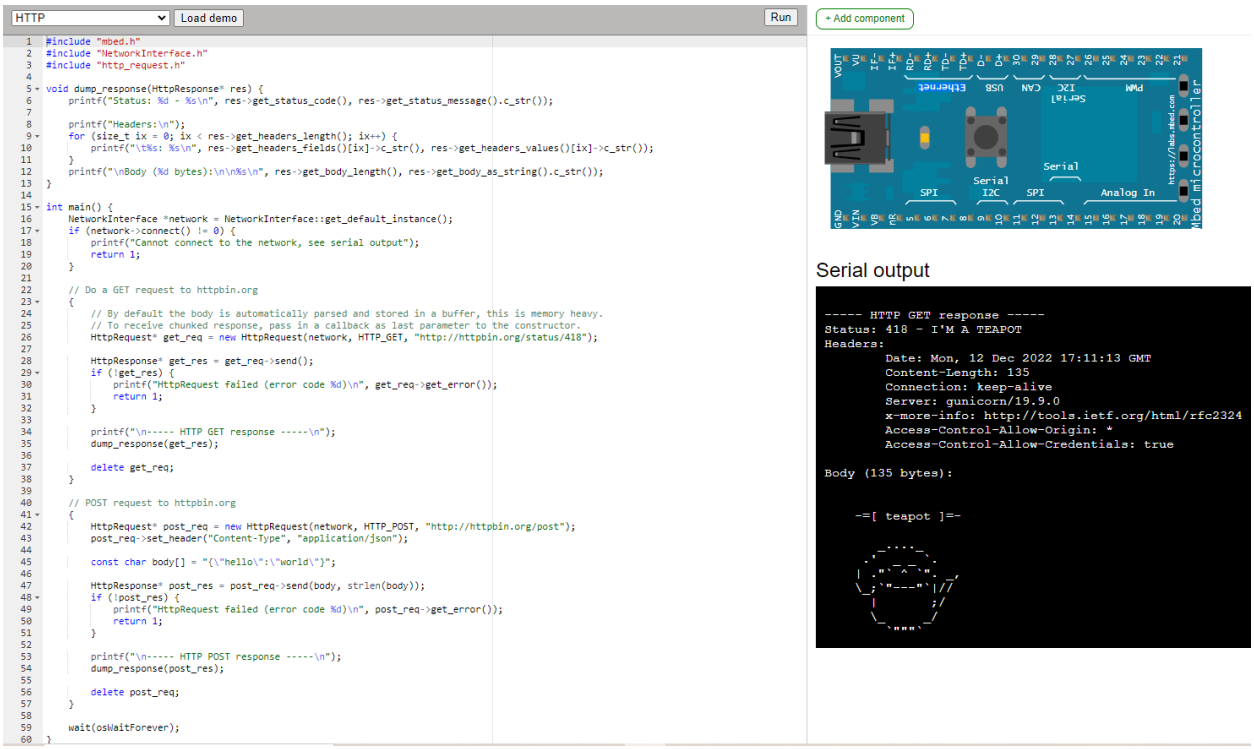
Περιέχουν πληροφορίες κειμένου αποθηκευμένες σε ζεύγη κλειδιών-τιμών και περιλαμβάνονται σε κάθε αίτηση HTTP. Αυτές οι επικεφαλίδες επικοινωνούν βασικές πληροφορίες όπως ποιο πρόγραμμα περιήγησης χρησιμοποιεί ο πελάτης και ποια δεδομένα ζητούνται.

HTTP response (κωδικός κατάστασης, επικεφαλίδες απόκρισης, προαιρετικό σώμα)

Μια απάντηση HTTP είναι αυτό που λαμβάνουν οι πελάτες ιστού από έναν διακομιστή διαδικτύου ως απάντηση σε ένα αίτημα HTTP.

Status Code (κωδικός κατάστασης)

Είναι τριψήφιοι κωδικοί που χρησιμοποιούνται για να δηλώσουν αν ένα αίτημα HTTP έχει ολοκληρωθεί επιτυχώς.



ΠΕΡΙΓΡΑΦΗ: Ανταλλαγή πληροφοριών-δεδομένων ανάμεσα σε πελάτη και διακομιστής. Πιο συγκεκριμένα, έχουμε δύο περιπτώσεις, στην πρώτη έχουμε τον κώδικα απόκρισης σφάλματος του 418 I'm teapot client που υποδεικνύει ότι ο διακομιστής αρνείται να ετοιμάσει καφέ επειδή είναι μια τσαγιέρα. Στην ουσία οποιαδήποτε προσπάθεια παρασκευής καφέ με μία τσαγιέρα θα πρέπει να οδηγεί στον κωδικό σφάλματος "418 I'm teapot".Ο κωδικός 418 χρησιμοποιείται επίσης για να επιστρέφει " I'm teapot" όταν ζητείται από τον διακομιστή να ετοιμάσει καφέ. Στην δεύτερη, επιστρέφει τις επικεφαλίδες HTTP της εισερχόμενης αίτησης.

COAP: Constrained Application Protocol (Πρωτόκολλο Περιορισμένης Εφαρμογής)

Εξειδικευμένο πρωτόκολλο εφαρμογών του διαδικτύου για περιορισμένες συσκευές που ονομάζονται κόμβοι να επικοινωνούν με το ευρύτερο διαδίκτυο χρησιμοποιώντας παρόμοια πρωτόκολλα.

Μορφές μηνυμάτων

Χρησιμοποιεί 2 μορφές μηνυμάτων : Requests(αιτήσεις), Responses(απαντήσεις) .

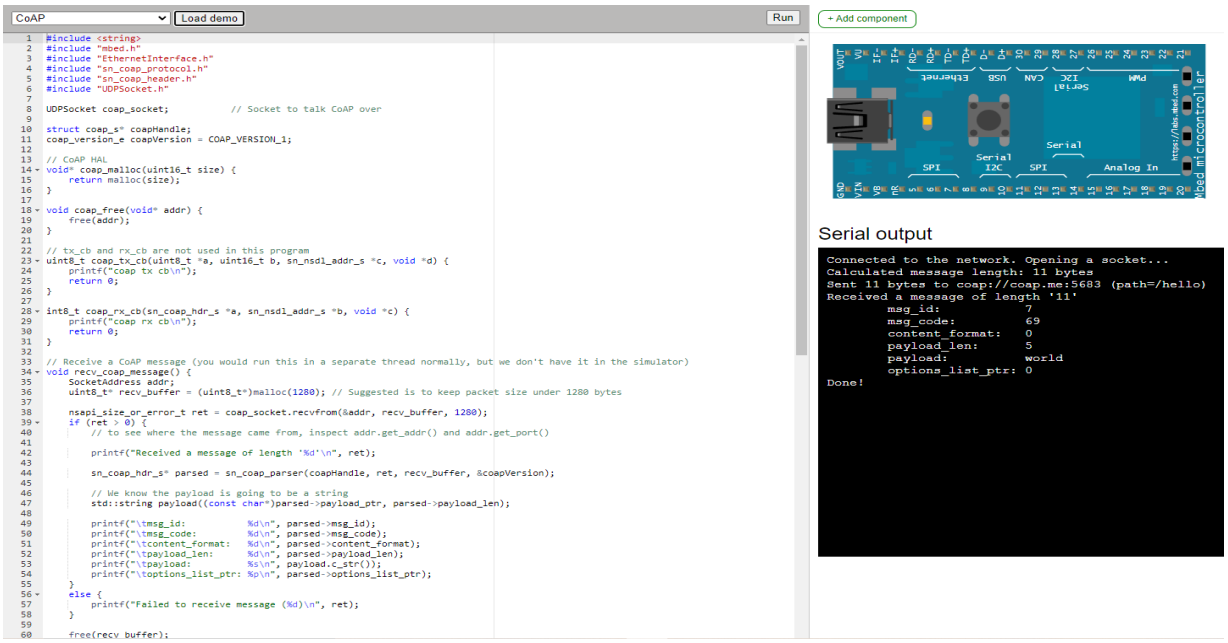
Τύπος μηνυμάτων

Αίτηση : -0 : Επιβεβαιώσιμο : Το μήνυμα αυτό αναμένει ένα αντίστοιχο μήνυμα επιβεβαίωσης.

-1: Μη επιβεβαιώσιμο : Το μήνυμα αυτό δεν αναμένει μήνυμα επιβεβαίωσης.

Απάντηση : -2 : Επιβεβαίωση : Το μήνυμα αυτό είναι μια απάντηση που επιβεβαιώνει ένα επιβεβαιώσιμο μήνυμα.

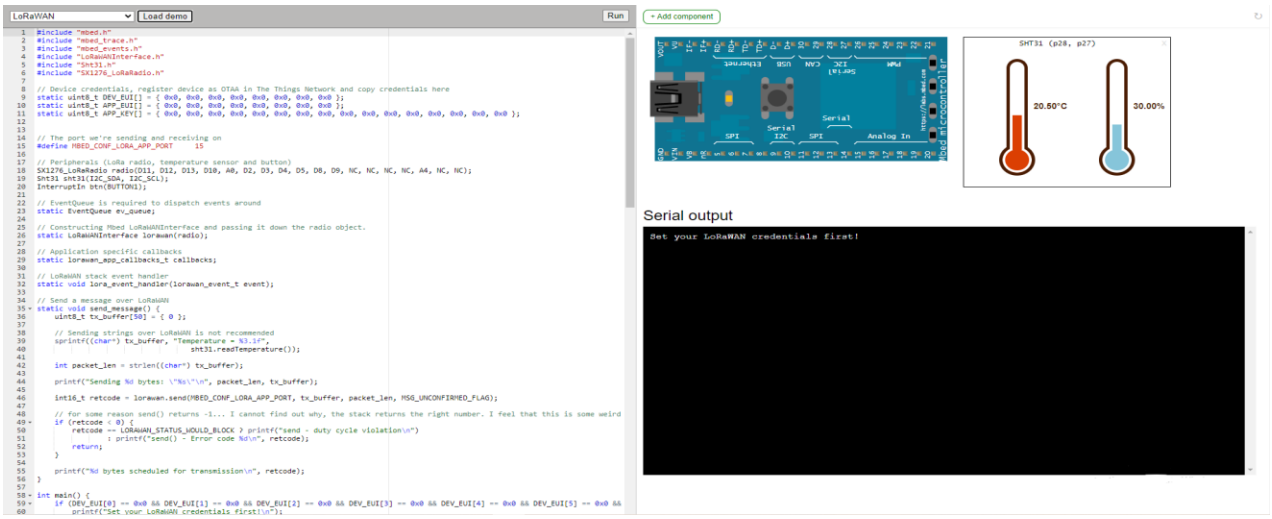
-3 : Επαναφορά : Το μήνυμα αυτό δηλώνει ότι έλαβε ένα μήνυμα αλλά δεν μπόρεσε να το επεξεργαστεί.



ΠΕΡΙΓΡΑΦΗ: Μας δείχνει πως να συνδεθούμε σε έναν διακομιστή Coap από το Embed. Συνδέεται στο coap://coap.me,στέλνει ένα μήνυμα στο /hello και λαμβάνει την απάντηση.

LORAWAN: Long Range Wide Area Network (Δίκτυο Ευρείας Περιοχής Μεγάλης Εμβέλειας)

Πρωτόκολλο επιπέδου ελέγχου πρόσβασης στο μέσο (MAC) βασισμένο στο cloud, αλλά λειτουργεί κυρίως ως πρωτόκολλο επιπέδου δικτύου για τη διαχείριση της επικοινωνίας μεταξύ των πυλών LPWAN και των συσκευών τελικού κόμβου ως πρωτόκολλο δρομολόγησης.



ΠΕΡΙΓΡΑΦΗ : Στέλνουμε την τιμή του αισθητήρα χρησιμοποιόντας την επικοινωνία LORA προς LORA. Ένα LORA ως πελάτης και ένα άλλο ως διακομιστής. Τα δεδομένα που στέλνουμε είναι η θερμοκρασία και η υγρασία και στην συνέχεια εμφανίζονται σε μία οθόνη Led στον διακομιστή.

Για να τρέξει η στοίβα Lora WAN,ο προσομοιωτής χρησιμοποιεί έναν ψεύτικο οδηγό ραδιοφώνου LoRa. Όταν η στοίβα Lora WAN θέλει να οδηγήσει το ραδιόφωνο, το ψεύτικο ραδιόφωνο αναχαιτίζει το πακέτο. Αυτά τα δεδομένα παραδίδονται στη συνέχεια απευθείας σε έναν διακομιστή δικτύου Lora WAN,ο οποίος δεν έχει ιδέα ότι δεν πρόκειται για μια πραγματική συσκευή Lora WAN.

TCP: Transmission Control Protocol (Πρωτόκολλο Ελέγχου Μετάδοσης)

Η υποδοχή TCP είναι μια υποδοχή προσανατολισμένη στη σύνδεση που χρησιμοποιεί το πρωτόκολλο ελέγχου μετάδοσης(TCP). Η κλάση TCPSocket παρέχει τη δυνατότητα αποστολής μιας ροής δεδομένων μέσω TCP. Διατηρούν μια κατάσταση σύνδεσης που ξεκινά με τη συνάρτηση μέλους connect. Χρησιμοποιείται για την επικοινωνία μεταξύ δύο υπολογιστών. Το socket περιλαμβάνει την διεύθυνση πρωτοκόλλου Internet(Ip),καθώς και τον κεντρικό υπολογιστή ή την θύρα που χρησιμοποιούν οι υπολογιστές για τη μετάδοση των δεδομένων. Απαιτεί τρία πακέτα για τη δημιουργία μιας σύνδεσης : το πακέτο SYN, SYN-ACK, ACK .

Στέλνοντας δεδομένα με το TCP Socket

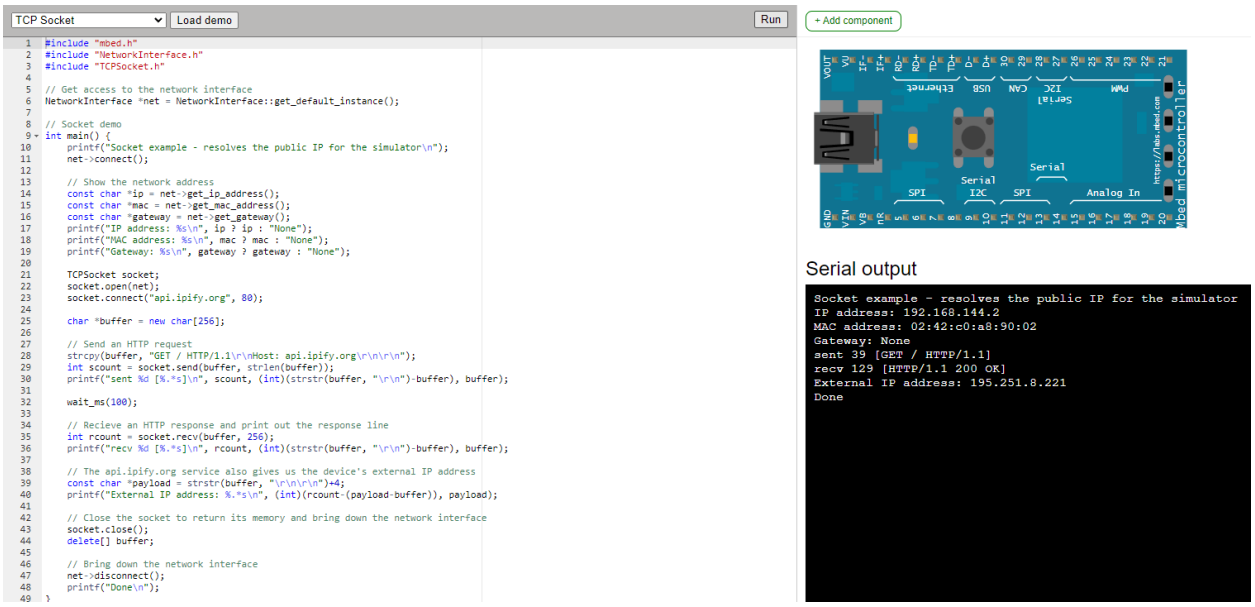
-Πρέπει πρώτα να δημιουργήσουμε μια σύνδεση πριν στείλουμε στην άλλη πλευρά.

-Παρέχει τις ρουτίνες που απαιτούνται για την επικοινωνία μεταξύ εφαρμογών, είτε στο τοπικό σύστημα είτε σε ένα κατανεμημένο δικτυακό περιβάλλον με βάση το TCP/IP.

Σκοπός

-Να προσδιορίζει τη μοναδική διεύθυνση IP του υπολογιστεί ή του διακομιστή δικτύου, μαζί με τον αριθμό θύρας του. Οι αριθμοί θυρών αναφέρονται μερικές φορές ως hosts. Μαζί η διεύθυνση IP και ο αριθμός θύρας συνθέτουν μια υποδοχή.

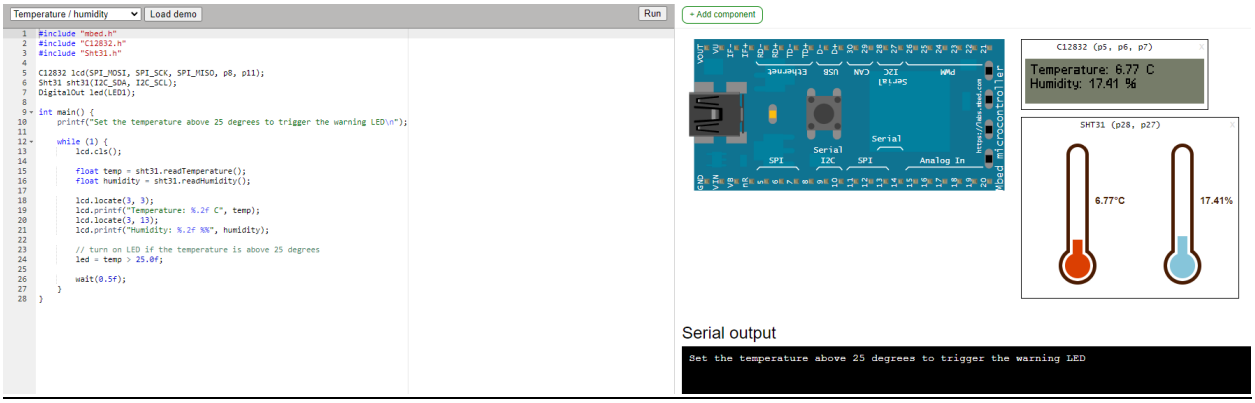
-Όταν δημιουργείται μια υποδοχή, τόσο ο αποστολέας όσο και ο παραλήπτης δοκιμάζουν την θύρα επικοινωνίας για να διασφαλίζουν ότι τα δεδομένα περνούν.



ΠΕΡΙΓΡΑΦΗ : Η υποδοχή διακομιστή παίρνει τα δεδομένα, τα ενθυλακώνει σε ένα πακέτο TCP και τα στέλνει στον απομακρυσμένο ομότιμο του. Στη συνέχεια, το TCP Socket διατηρεί τα αποσταλμένα πακέτα στη μνήμη και περιμένει μια επιβεβαίωση από τον απομακρυσμένο ομότιμο του. Εάν το πακέτο δεν επιβεβαιωθεί όταν λήξει το χρονικό όριο, το ίδιο πακέτο αποστέλλεται εκ νέου.

Temperature/humidity

Εξομοίωση του αισθητήρα θερμοκρασίας - υγρασίας Sensor(SHT31) βασισμένο στο εργαλείο embed simulator της ARM (https://simulator.mbed.com/).



ΠΕΡΙΓΡΑΦΗ : Στο παραπάνω παράδειγμα μέσω του αισθητήρα SHT 31 παραμετροποιούμε την θερμοκρασία όσες φορές θέλουμε και μας εμφανίζει τα αντίστοιχα δεδομένα θερμοκρασίας-υγρασίας στην οθόνη C12832.

MQTT: Message Queuing Telemetry Transport (Ουρά μηνυμάτων Μεταφοράς τηλεμετρίας)

-Το MQTT είναι ένα πρωτόκολλο ανταλλαγής μηνυμάτων που βασίζεται σε πρότυπα ή ένα σύνολο κανόνων, που χρησιμοποιείται για επικοινωνία από μηχανή με μηχανή. Οι συσκευές Internet of Things (IoT) πρέπει συνήθως να μεταδίδουν και να λαμβάνουν δεδομένα μέσω ενός δικτύου περιορισμένων πόρων με περιορισμένο εύρος ζώνης. Αυτές οι συσκευές IoT χρησιμοποιούν MQTT για μετάδοση δεδομένων, καθώς είναι εύκολο να εφαρμοστούν και μπορούν να επικοινωνήσουν δεδομένα IoT αποτελεσματικά. Το MQTT υποστηρίζει την ανταλλαγή μηνυμάτων μεταξύ συσκευών στο cloud και από το cloud στη συσκευή.

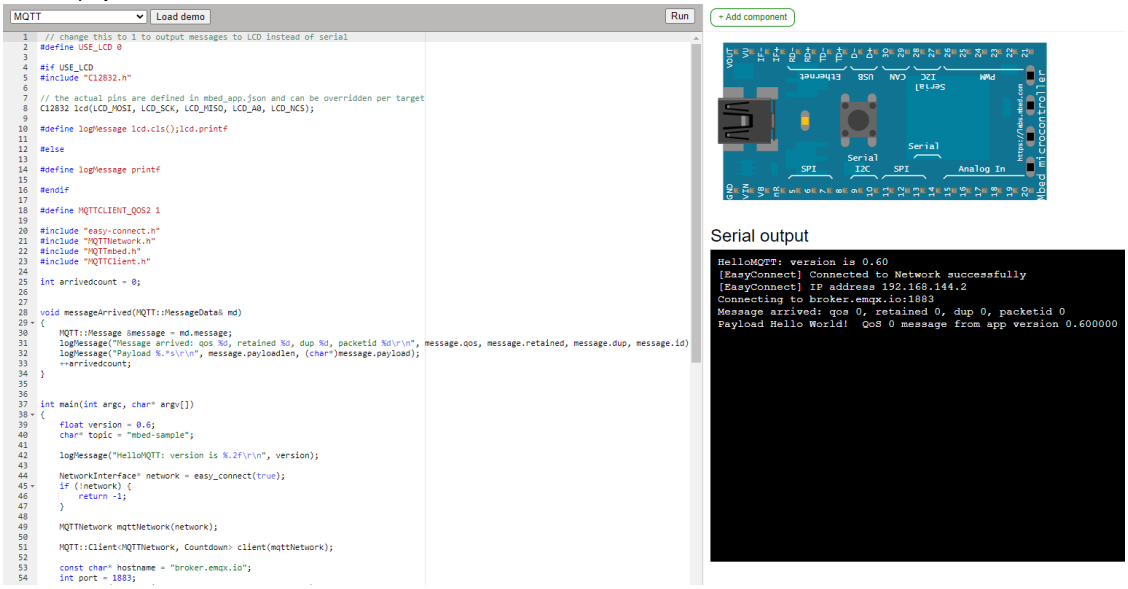
-Παρέχει σε πελάτες δικτύου με περιορισμένους πόρους έναν απλό τρόπο διανομής πληροφοριών τηλεμετρίας σε περιβάλλοντα χαμηλού εύρους ζώνης. Το πρωτόκολλο, το οποίο χρησιμοποιεί ένα μοτίβο επικοινωνίας δημοσίευσης/συνδρομής, χρησιμοποιείται για επικοινωνία μηχανής με μηχανή (M2M).

ΛΕΙΤΟΥΡΓΙΑ

1: Ένας client MQTT δημιουργεί μια σύνδεση με τον broker MQTT.

2: Μόλις συνδεθεί, ο client μπορεί είτε να δημοσιεύσει(publish) μηνύματα, να εγγραφεί(subscribe) σε συγκεκριμένα μηνύματα ή να κάνει και τα δύο.

3: Όταν ο broker MQTT λαμβάνει ένα μήνυμα, το προωθεί στους συνδρομητές(subscribers) που ενδιαφέρονται.



ΠΕΡΙΓΡΑΦΗ : Αυτό είναι ένα πρόγραμμα-δείγμα για την παρουσίαση της χρήσης της βιβλιοθήκης MQTT Client στην πλατφόρμα mbed. Η κλάση Client απαιτεί δύο κλάσεις οι οποίες μεσολαβούν στην πρόσβαση στις διεπαφές του συστήματος για τη δικτύωση και το χρονισμό. Εφόσον αυτές οι δύο κλάσεις παρέχουν τις απαιτούμενες δημόσιες προγραμματιστικές διεπαφές, δεν έχει σημασία ποιες διευκολύνσεις χρησιμοποιούν από κάτω. Σε αυτό το πρόγραμμα, χρησιμοποιούν τις βιβλιοθήκες συστήματος mbed. Το HelloMQTT είναι ένα παράδειγμα χρήσης του MQTT API. Το MQTT API είναι φορητό σε στοίβες διεπαφής δικτύου. Το MQTT έχει σχεδιαστεί για χρήση με TCP/IP, αλλά οποιαδήποτε μεταφορά με παρόμοια χαρακτηριστικά θα πρέπει να είναι κατάλληλη.

Κεφάλαιο 5 : Παρουσίαση Demo

Στο παρόν Demo θα ήθελα να σας παρουσιάσω ένα πρόγραμμα βασισμένο στο πρωτόκολλο επικοινωνίας MQTT που εκτελείται με βάση τον Mbed simulator και σε συνεργασία με την εφαρμογή MqttBox η οποία δέχεται τα δεδομένα με την μορφή δημοσίευσης (publish) μέσω του component SHT31 sensor και στην συνέχεια τα στέλνει πίσω με την μορφή εγγραφής (subscribe). Θα σταθούμε πιο λεπτομερώς και στην συνέχεια καθώς για αρχή ας δούμε με πιο σαφήνεια το πρωτόκολλο Mqtt καθώς και την εφαρμογή MqttBox την οποία και θα την χρησιμοποιήσουμε όπως αναφέρθηκα και πιο πάνω για την υλοποίηση του Demo.

Ας ξεκινήσουμε λοιπόν με τον ορισμό του πρωτοκόλλου Mqtt, το MQTT (Message Queuing Telemetry Transport-Μεταφορά τηλεμετρίας ουράς μηνυμάτων) είναι ένα πρωτόκολλο ανταλλαγής μηνυμάτων που βασίζεται σε πρότυπα ή ένα σύνολο κανόνων, που χρησιμοποιείται για επικοινωνία από μηχανή με μηχανή. Οι συσκευές Internet of Things (IoT) πρέπει συνήθως να μεταδίδουν και να λαμβάνουν δεδομένα μέσω ενός δικτύου περιορισμένων πόρων με περιορισμένο εύρος ζώνης. Αυτές οι συσκευές IoT χρησιμοποιούν MQTT για μετάδοση δεδομένων, καθώς είναι εύκολο να εφαρμοστούν και μπορούν να επικοινωνήσουν δεδομένα IoT αποτελεσματικά. Το MQTT υποστηρίζει την ανταλλαγή μηνυμάτων μεταξύ συσκευών στο cloud και από το cloud στη συσκευή.

Παρέχει σε πελάτες δικτύου με περιορισμένους πόρους έναν απλό τρόπο διανομής πληροφοριών τηλεμετρίας σε περιβάλλοντα χαμηλού εύρους ζώνης. Το πρωτόκολλο, το οποίο χρησιμοποιεί ένα μοτίβο επικοινωνίας δημοσίευσης/συνδρομής, χρησιμοποιείται για επικοινωνία μηχανής με μηχανή (M2M).

ΛΕΙΤΟΥΡΓΙΑ

1: Ένας client MQTT δημιουργεί μια σύνδεση με τον broker MQTT.

2: Μόλις συνδεθεί, ο client μπορεί είτε να δημοσιεύσει(publish) μηνύματα, να εγγραφεί(subscribe) σε συγκεκριμένα μηνύματα ή να κάνει και τα δύο.

3: Όταν ο broker MQTT λαμβάνει ένα μήνυμα, το προωθεί στους συνδρομητές(subscribers) που ενδιαφέρονται.

-Όταν μια συσκευή (ή πελάτης) θέλει να στείλει δεδομένα σε διακομιστή (ή broker) ονομάζεται δημοσίευση .

-Όταν η λειτουργία αντιστρέφεται, ονομάζεται συνδρομή .

-Ο Server είναι ο broker, ενώ οι συνδεδεμένες συσκευές είναι οι clients.

MQTT Client (πελάτης)

- ✓ Ένας client MQTT είναι οποιαδήποτε συσκευή από διακομιστή έως μικροελεκτή που εκτελεί μια βιβλιοθήκη MQTT. Εάν ο πελάτης στέλνει μηνύματα, ενεργεί ως Publisher και εάν λαμβάνει μηνύματα, ενεργεί ως παραλήπτης. Βασικά, οποιαδήποτε συσκευή που επικοινωνεί χρησιμοποιώντας MQTT μέσω δικτύου μπορεί να ονομαστεί συσκευή πελάτη MQTT.
- ✓ Ένας πελάτης μπορεί να στέλνει μηνύματα στον broker (publish) και να λαμβάνει μηνύματα από τον broker (subscribe). Κατά την αποστολή ενός μηνύματος στον broker, πρέπει να καθοριστεί ένα θέμα MQTT, το οποίο μπορεί να χρησιμοποιηθεί για την περαιτέρω επεξεργασία του μηνύματος.
- ✓ Στο MQTT, ο client εκτελεί δύο λειτουργίες :

    1. Δημοσίευση(Publish): Όταν ο client στέλνει τα δεδομένα στον Server, τότε ονομάζουμε αυτή τη λειτουργία ως δημοσίευση.
    2. Εγγραφή(Subscribe): Όταν ο client λάβει τα δεδομένα από τον Server, τότε ονομάζουμε αυτή τη λειτουργία συνδρομή.

- ✓ Οι client MQTT δημοσιεύουν ένα μήνυμα σε έναν broker MQTT και άλλοι clients MQTT εγγράφονται σε μηνύματα που θέλουν να λαμβάνουν.

MQTT Topic (Θέμα)

- Ο όρος «θέμα» αναφέρεται σε λέξεις-κλειδιά που χρησιμοποιεί ο μεσίτης MQTT για να φιλτράρει μηνύματα για τους πελάτες MQTT. Τα θέματα οργανώνονται ιεραρχικά, παρόμοια με έναν κατάλογο αρχείων ή φακέλων.
- Είναι ο τρόπος με τον οποίο δηλώνετε ενδιαφέρον για τα εισερχόμενα μηνύματα ή πώς προσδιορίζετε πού θέλετε να δημοσιεύσετε το μήνυμα.
- Τα θέματα αντιπροσωπεύονται με συμβολοσειρές που χωρίζονται με κάθετο προς τα εμπρός. Κάθε κάθετο προς τα εμπρός υποδεικνύει
- ένα επίπεδο θέματος. Ακολουθεί ένα παράδειγμα για το πώς θα δημιουργήσατε ένα θέμα για μια λάμπα στο γραφείο του σπιτιού σας :

home/office/lamp
topic
level
/: topic level separator (διαχωριστικό θεματικού επιπέδου)

MQTT Broker(μεσίτης)

- Ο Broker MQTT είναι το κέντρο κάθε πρωτοκόλλου Publish/Subscribe. Ανάλογα με την υλοποίηση, ένας μεσίτης μπορεί να διαχειριστεί έως και χιλιάδες ταυτόχρονα συνδεδεμένους πελάτες MQTT.Ο Broker είναι υπεύθυνος για τη λήψη όλων των μηνυμάτων, το φιλτράρισμα των μηνυμάτων, τον προσδιορισμό του ποιος εγγράφηκε σε κάθε μήνυμα και την αποστολή του μηνύματος σε αυτούς τους εγγεγραμμένους πελάτες.
- Ο έλεγχος ταυτότητας και η εξουσιοδότηση πελατών.
- Είναι ο κεντρικός κόμβος μέσω του οποίου πρέπει να δρομολογηθεί κάθε μήνυμα.
- Ο Broker MQTT είναι το σύστημα υποστήριξης που συντονίζει τα μηνύματα μεταξύ των διαφορετικών πελατών.

MQTT PAYLOAD(ωφέλιμο φορτίο)

- Κάθε μήνυμα περιέχει ένα περιεχόμενο μηνύματος , το λεγόμενο ωφέλιμο φορτίο.
- Το ωφέλιμο φορτίο MQTT δεν δεσμεύεται μια συγκεκριμένη δομή και μπορεί να σχεδιαστεί ελεύθερα.

Quality of Service (QoS)(Επίπεδα Ποιότητας Υπηρεσίας)

- Το QoS αναφέρεται σε μια συμφωνία μεταξύ του αποστολέα ενός μηνύματος και του παραλήπτη του μηνύματος. Λειτουργεί ως βασικό χαρακτηριστικό στο MQTT, δίνοντας στον πελάτη τη δυνατότητα να επιλέξει μεταξύ τριών επιπέδων υπηρεσίας.
- Το MQTT εφαρμόζει 3 επίπεδα ποιότητας υπηρεσιών για συμφωνία μεταξύ του αποστολέα και του παραλήπτη.

  - **QoS 0** : Το μήνυμα του client αποστέλλεται ακριβώς μία φορά, ανεξάρτητα από το αν έχει φτάσει στον broker.
  - **QoS 1** : Το μήνυμα του client αποστέλλεται ξανά και ξανά έως ότου ο broker απαντήσει με επιβεβαίωση παραλαβής. Αυτό μπορεί να έχει ως αποτέλεσμα την άφιξη ενός μηνύματος στον broker πολλές φορές.
  - **QoS 2** : Ο client στέλνει ένα μήνυμα μία φορά και ταυτόχρονα διασφαλίζει ότι έχει φτάσει στον broker. Η επικοινωνία Quality of Service 2 απαιτεί περισσότερο εύρος ζώνης από την Quality of Service 0 ή 1.

Connection MQTT(ΣΥΝΔΕΣΗ)

Οι clients και οι brokers αρχίζουν να επικοινωνούν χρησιμοποιώντας μια σύνδεση MQTT. Οι clients ξεκινούν τη σύνδεση στέλνοντας ένα μήνυμα CONNECT στον broker MQTT. Ο broker επιβεβαιώνει ότι έχει δημιουργηθεί μια σύνδεση απαντώντας με ένα μήνυμα CONNACK .Τόσο ο client MQTT όσο και ο broker απαιτούν μια στοίβα TCP/IP για να επικοινωνήσουν. Οι clients δεν συνδέονται ποτέ μεταξύ τους, μόνο με τον broker.

MQTT PUBLISH(ΔΗΜΟΣΙΕΥΣΗ)

- Οι clients MQTT δημοσιεύουν μηνύματα που περιέχουν το θέμα και τα δεδομένα σε μορφή byte. Ο clients καθορίζει τη μορφή δεδομένων, όπως δεδομένα κειμένου, δυαδικά δεδομένα, αρχεία XML ή JSON.
- Ένας client χρησιμοποιεί αυτήν την εντολή για να στείλει μηνύματα στον broker.

MQTT SUBSCRIBE(ΕΓΓΡΑΦΗ)

- Οι πελάτες MQTT στέλνουν ένα μήνυμα SUBSCRIBE στον broker MQTT, για να λάβουν μηνύματα για θέματα που σας ενδιαφέρουν. Αυτό το μήνυμα περιέχει ένα μοναδικό αναγνωριστικό και μια λίστα συνδρομών.
- Ο πελάτης το χρησιμοποιεί αυτό για να λαμβάνει μηνύματα από τον μεσίτη και μετατρέπει έναν πελάτη σε συνδρομητή θέματος.

MQTT MESSAGES(ΜΗΝΥΜΑΤΑ)

- CONNECT: αίτηση client να συνδεθεί στον server (τιμή :1 - κατεύθυνση: client to server)
- CONNACK (CONNECT ACKNOWLADGEMENT-ΕΠΙΒΕΒΑΊΩΣΗ ΣΥΝΔΕΣΗΣ): (τιμή :2 – κατεύθυνση: server to client)
- PUBLISH: δημοσίευση μηνύματος (τιμή: 3 – κατεύθυνση: client to server ή server to client)
- PUBACK (PUBLISH ACKNOWLADGEMENT-ΕΠΙΒΕΒΑΊΩΣΗ ΔΗΜΟΣΙΕΥΣΗΣ): (τιμή: 4 – κατεύθυνση: client to server ή server to client)
- PUBREC (PUBLISH RECEIVED-ΔΗΜΟΣΊΕΥΣΗ ΠΟΥ ΈΛΑΒΕ): (τιμή: 5 – κατεύθυνση: client to server ή server to client)
- PUBREL (PUBLISH RELEASE-ΔΗΜΟΣΙΕΎΣΤΕ ΤΗΝ ΈΚΔΟΣΗ): (τιμή: 6 – κατεύθυνση: client to server ή server to client)
- PUBCOMP (PUBLISH COMPLETE – Η Δημοσίευση ολοκληρώθηκε): (τιμή: 7 – κατεύθυνση: client to server ή server to client)
- SUBSCRIBE: ο client κάνει εγγραφή αίτησης (τιμή :8 - κατεύθυνση: client to server)

- SUBACK (SUBSCRIBE ACKNOWLADGEMENT-ΕΠΙΒΕΒΑΊΩΣΗ εγγραφής): (τιμή: 9 – κατεύθυνση: server to client)
- UNSUBSCRIBE: απεγγραφή αίτησης (τιμή: 10 - κατεύθυνση: client to server)
- UNSUBACK (UNSUBSCRIBE ACKNOWLADGEMENT-ΕΠΙΒΕΒΑΊΩΣΗ απεγγραφής): (τιμή :11 – κατεύθυνση: server to client)
- PINGREQ (PING REQUEST- αίτηση ping): (τιμή :12 - κατεύθυνση: client to server)
- PINGRSP (PING RESPONSE- ΑΠΑΝΤΗΣΗ PING): (τιμή :13 - κατεύθυνση: server to client)

MqttBox

Βοηθητικό πρόγραμμα προγραμματιστών για τη δημιουργία, την ανάπτυξη και τη δοκιμή του πρωτοκόλλου συνδεσιμότητας MQTT. Το MQTTBox μας επιτρέπει να δημιουργούμε πελάτες MQTT για να δημοσιεύουμε(publish) ή να εγγράφουμε(subscribe) σε topics, να δημιουργείτε virtual devices MQTT, να δοκιμάζετε συσκευές ή brokers MQTT και πολλά άλλα.

MQTTbox έχει δύο μορφές :

1. Το πρόσθετο στο πρόγραμμα περιήγησης chrome.
2. Την εφαρμογή nodejs App.

Το MQTTBox είναι ένα εργαλείο για προγραμματιστές που κάνει την παρακολούθηση του Server MQTT πανεύκολη. Μπορούμε να εγγραφούμε(subscribe) και να δημοσιεύσουμε(publish) topics χρησιμοποιώντας το MQTTBox, καθώς και να δημιουργήσουμε ένα virtual device MQTT για δοκιμές και αποσφαλμάτωση. Το καλύτερο μέρος είναι ότι μπορούμε να χρησιμοποιήσουμε τον ίδιο διακομιστή για πολλαπλούς σκοπούς, ώστε να αξιοποιήσουμε στο έπακρο τη ροή εργασίας της ανάπτυξής μας.

Αρχικά χρησιμοποιήθηκε μόνο ως εκτεταμένη εγκατάσταση στο Chrome, και αργότερα ξαναγράφτηκε και αποκτήθηκε με ανοιχτό κώδικα για να γίνει ένα ανεξάρτητο λογισμικό για επιτραπέζια επιφάνεια εργασίας με πολλαπλές πλατφόρμες. Το περιβάλλον εργασίας είναι απλό και ξεκάθαρο και υποστηρίζει πολλούς clients σε απευθείας σύνδεση ταυτόχρονα. Ωστόσο, εξακολουθούν να υπάρχουν ορισμένες δυσχέρειες στην αλληλεπίδραση μεταξύ των clients, όπως η εναλλαγή και η αποστολή μηνυμάτων. Το MQTTBox επιτυγχάνει ισχυρά χαρακτηριστικά cross-platform με το Chrome. Σε συνδυασμό με απλές λειτουργίες δοκιμής φορτίου, είναι ένα εργαλείο πελάτη MQTT που αξίζει να δοκιμάσετε.

MQTTBox clients:

- ✓ Δημιουργία πολλαπλών ταυτόχρονων clients MQTT που συνδέονται με τους ίδιους ή διαφορετικούς brokers MQTT.
- ✓ Υποστήριξη TCP, SSL/TLS, Web Sockets και Secured Web Sockets.

- ✓ Δημιουργούμαι clients MQTT με ευρύ φάσμα ρυθμίσεων σύνδεσης - Last will, keepalive, χρονικό όριο σύνδεσης και πολλά άλλα .
- ✓ Αυθεντικοποίηση με username/password.
- ✓ Δυνατότητα σύνδεσης, αποσύνδεσης, επανασύνδεσης του client MQTT με τους brokers.
- ✓ Δημοσίευση(Publish) μηνυμάτων σε πολλαπλά topics από τον ίδιο client MQTT.
- ✓ Subscribe σε πολλαπλά topics από τον ίδιο client MQTT.
- ✓ Υποστηρίζει μονοεπίπεδη(+) και πολυεπίπεδη(#) συνδρομή σε topics.
- ✓ Αντιγραφή, αναδημοσίευση ωφέλιμου φορτίου με ένα μόνο κλικ.
- ✓ Προβολή ιστορικού δημοσιευμένων και εγγεγραμμένων μηνυμάτων.
- ✓ Υποστηρίζει QoS 0,1,2 .

MQTTBox load

- ✓ Δοκιμή φορτίου υποδομής MQTT - συσκευές MQTT, διαμεσολαβητές, cloud και εφαρμογές.
- ✓ Δημιουργία πολλαπλών περιπτώσεων δοκιμής φορτίου.
- ✓ Δημιουργήστε δοκιμή φορτίου με μεγάλο εύρος ρυθμίσεων - αριθμός μηνυμάτων για publish/subscribe ανά δευτερόλεπτο, αριθμός στιγμιότυπων που πρέπει να καλύπτει .
- ✓ Δοκιμή φόρτωσης MQTT publish σε topics.
- ✓ Δοκιμή φορτίου MQTT subscribe σε topics.
- ✓ Προβολή της προόδου σε πραγματικό χρόνο - σύνδεση, αποσύνδεση, δημοσίευση, εγγραφή, λαμβανόμενα μηνύματα κ.λπ.
- ✓ Προβολή των αποτελεσμάτων της δοκιμής φορτίου σε διαγράμματα.
- ✓ Προβολή δεδομένων δοκιμής φορτίου που αποστέλλονται ή λαμβάνονται.
- ✓ Υπολογισμός του αριθμού των διακοπών σύνδεσης.

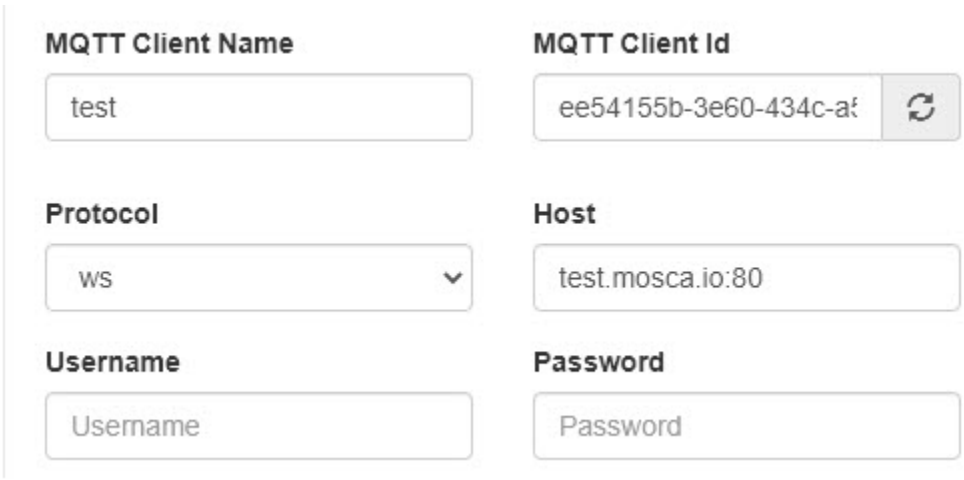
Το περιβάλλον του MqttBox

Το πρώτο βήμα είναι η δημιουργία μιας νέας σύνδεσης πελάτη. Για να το κάνουμε αυτό, ανοίγουμε την οθόνη MQTT clients χρησιμοποιώντας το μενού πάνω αριστερά.

Η οθόνη εμφανίζει τους υπάρχοντες πελάτες, αν υπάρχουν, και την κατάσταση σύνδεσής τους (πράσινο (συνδεδεμένοι) ή κόκκινο (αποσυνδεδεμένοι),

Κάντε κλικ στο μπλε κουμπί create mqtt client για να δημιουργήσετε έναν νέο πελάτη. Ανοίγει ένα παράθυρο διαλόγου και πρέπει να εισαγάγετε τα στοιχεία του πελάτη. Υποχρεωτικά είναι τα εξής:

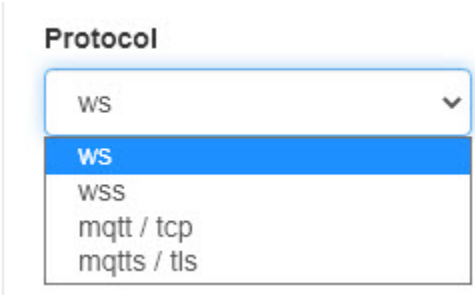
- Θύρα υποδοχής και IP (Host port and IP)
- Πρωτόκολλο (Protocol)
- Όνομα πελάτη (Client Name)



Παρατηρήστε ότι η διεύθυνση ip και η θύρα ρυθμίζονται στο πλαίσιο host.

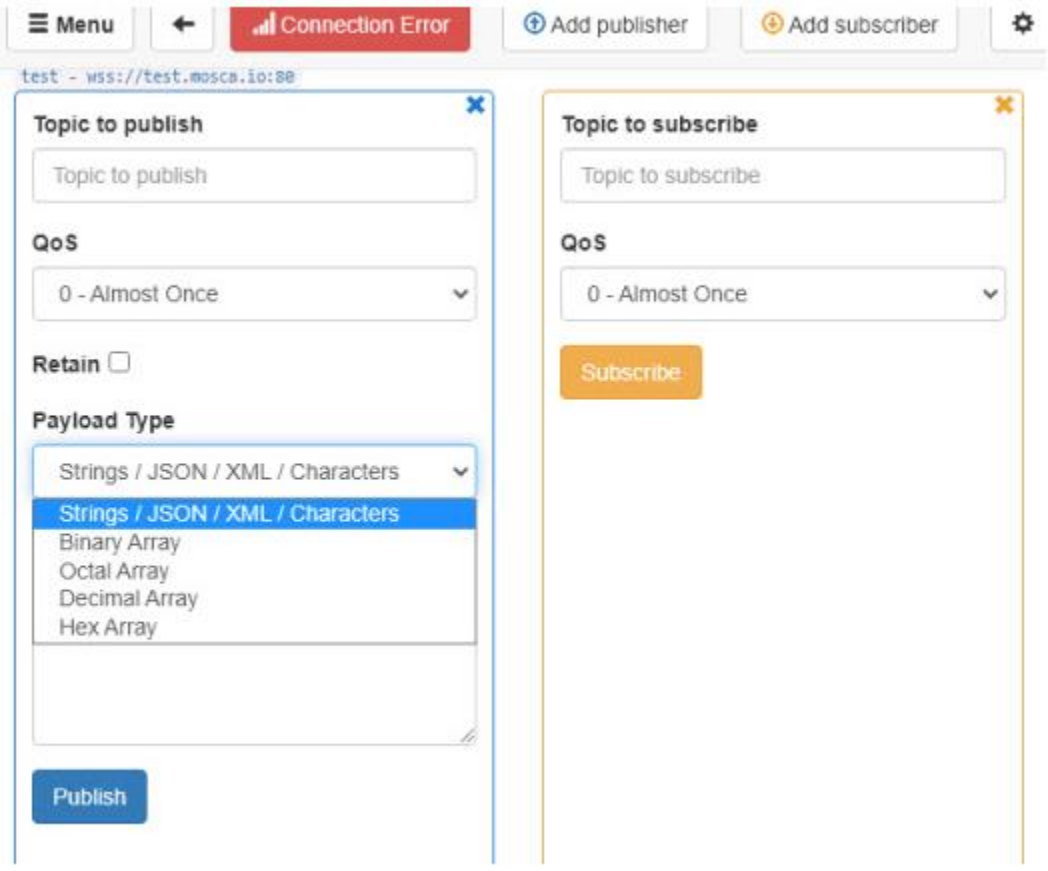
Τα πιθανά πρωτόκολλα είναι :

- MQTT μέσω TCP
- MQTT μέσω TCP με SSL (μόνο έκδοση APP)
- MQTT μέσω Websockets
- MQTT μέσω Websockets με SSL



Ο πελάτης υποστηρίζει τα βασικά χαρακτηριστικά του MQTT v3.1 θα μηνύματα, καθαρές συνεδρίες, διατηρημένα μηνύματα και επίπεδα QOS 0,1,2.

Όταν αποθηκεύσουμε τις ρυθμίσεις σύνδεσης, εμφανίζονται οι επιλογές δημοσίευσης και εγγραφής.



Η πιο συνηθισμένη μορφή για τα δημοσιευμένα μηνύματα είναι η συμβολοσειρά, η οποία περιλαμβάνει επίσης δεδομένα JSON.

Μπορούμε επίσης να ελέγξουμε τη σύνδεση χρησιμοποιώντας το κουμπί συνδεδεμένο στο επάνω μέρος του παραθύρου της εφαρμογής.

Απενεργοποιεί/ενεργοποιεί τη σύνδεση.

EMQX Cloud

Ο broker που θα χρησιμοποιήσουμε για την υλοποίηση του Demo είναι ο EMQX. Ο EMQX Cloud είναι μια πλήρως διαχειριζόμενη cloud-native υπηρεσία MQTT που εγκαινιάστηκε από την εταιρεία EMQ, η οποία μπορεί να συνδεθεί σε μεγάλο αριθμό συσκευών IoT και να ενσωματώσει διάφορες βάσεις δεδομένων και επιχειρηματικά συστήματα. Παρέχει υπηρεσία μηνυμάτων MQTT με διαχείριση λειτουργίας και συντήρησης από μία στάση και ένα μοναδικό περιβάλλον απομόνωσης. Στην εποχή του Internet of Everything, μπορεί να βοηθήσει τους χρήστες να δημιουργήσουν γρήγορα βιομηχανικές εφαρμογές για την περιοχή IoT και να πραγματοποιήσουν εύκολα τη συλλογή, τη μετάδοση, τον υπολογισμό και τη διατήρηση δεδομένων IoT.

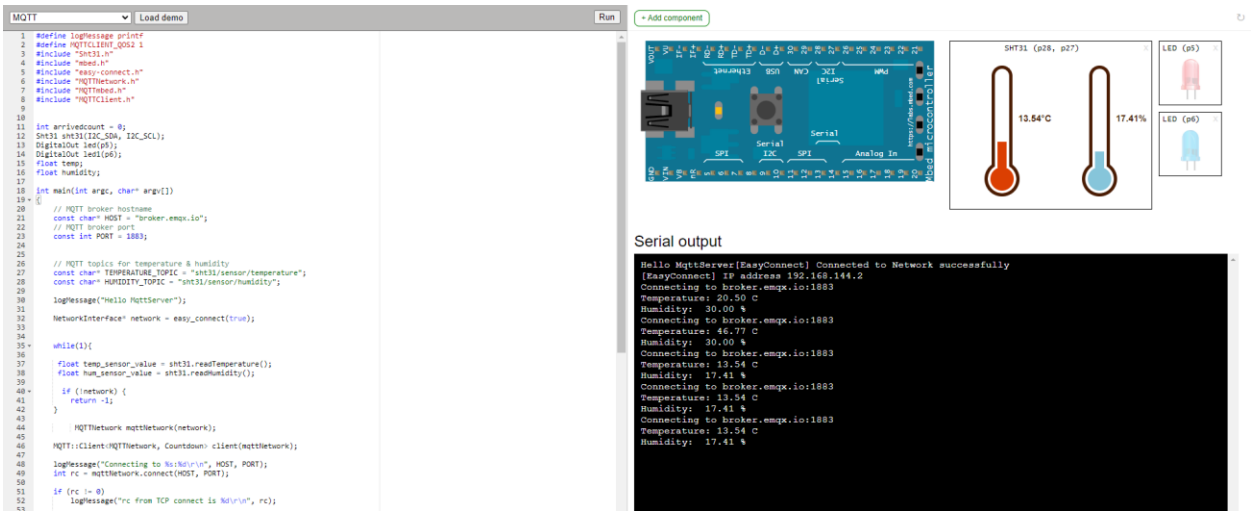
DEMO

Όπως αναφέρθηκα και πιο πάνω σε αυτό το demo θα δούμε μια προσομοίωση πάνω στο πρωτόκολλο επικοινωνίας MQTT με βάση την online πλατφόρμα του mbed Simulator.

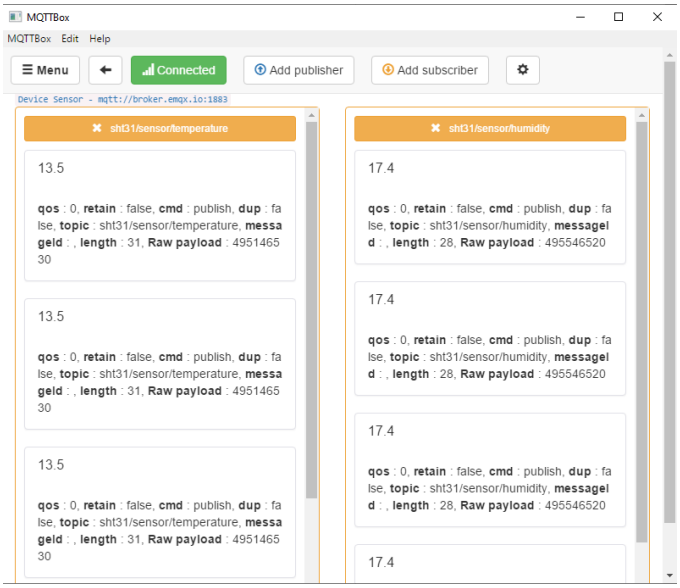
- Πιο συγκεκριμένα, θα υλοποιηθούν 2 λειτουργίες mqtt για την υλοποίηση του προγράμματος, η πρώτη θα έχει να κάνει με την λειτουργία publish(δημοσίευση) στην οποία με την ενσωμάτωση component, τον αισθητήρα θερμοκρασίας - υγρασίας sht31 θα στέλνουμε τις τιμές του αισθητήρα στην εφαρμογή MqttBox, όπου και η οποία θα μας βοηθάει καθόλου την διάρκεια της εκτέλεσης του προγράμματος να βλέπουμε τις τιμές της θερμοκρασίας ,της υγρασίας ακόμα και μερικές άλλες πληροφορίες όπως το qos(Quality of Service), retain(ο Publisher μπορεί να πει στον broker να κρατήσει το τελευταίο μήνυμα σε αυτό το topic), cmd-command(μας δείχνει το είδος της λειτουργίας που εκτελείται), dup-duplicate(υποδεικνύει ότι το μήνυμα είναι διπλότυπο και εστάλη εκ νέου επειδή ο προοριζόμενος παραλήπτης δεν επιβεβαίωσε το αρχικό μήνυμα.),topic(ένα

φίλτρο που χρησιμοποιεί ο broker στις παραδόσεις μηνυμάτων MQTT), messageId(μας δείχνει την id του μηνύματος), lenght (το μήκος του μηνύματος) και raw payload (το ανεπεξέργαστο ωφέλιμο φορτίο) κατά το χρονικό περιθώριο που έχουμε ορίσει. Επιπλέον όμως έχουμε δηλώσει και 2 led, ένα κόκκινο για την θερμοκρασία και ένα μπλε για την υγρασία τα οποία ανάλογα με το όριο που έχουμε ορίσει θα ανοίγουν και θα κλείνουν.

Το Interface του simulator για την λειτουργία publish



Το Interface του MqttBox



Main Code:

```
#define logMessage printf
#define MQTTCLIENT_QOS2 1
#include "Sht31.h"
#include "mbed.h"
#include "easy-connect.h"
#include "MQTTNetwork.h"
#include "MQTTmbed.h"
#include "MQTTClient.h"


int arrivedcount = 0;
Sht31 sht31(I2C_SDA, I2C_SCL);
DigitalOut led(p5);
DigitalOut led1(p6);
float temp;
float humidity;

int main (int argc, char* argv [])
{
    // MQTT broker hostname
    const char* HOST = "broker.emqx.io";
    // MQTT broker port
    const int PORT = 1883;


    // MQTT topics for temperature & humidity
```

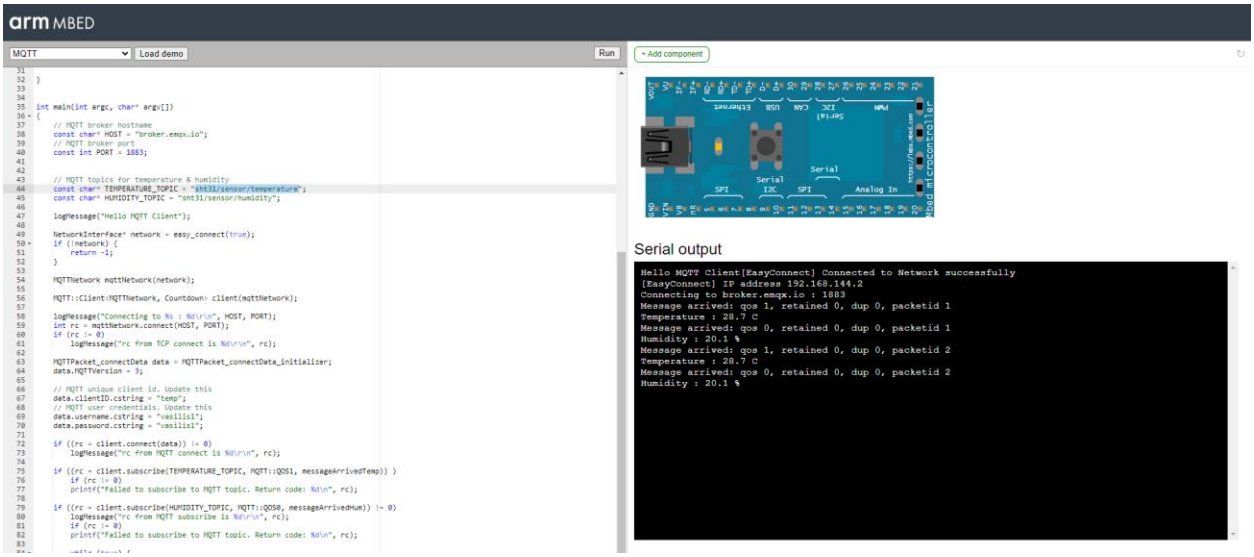
```
const char* TEMPERATURE_TOPIC = "sht31/sensor/temperature";
const char* HUMIDITY_TOPIC = "sht31/sensor/humidity";

logMessage ("Hello MqttServer");

NetworkInterface* network = easy_connect(true);



while (1) {

 float temp_sensor_value = sht31.readTemperature();
 float hum_sensor_value = sht31.readHumidity();

  if (! network) {
    return -1;
}

     MQTTNetwork mqttNetwork(network);

MQTT: Client<MQTTNetwork, Countdown> client(mqttNetwork);

logMessage ("Connecting to %s: %d\r\n", HOST, PORT);
int rc = mqttNetwork.connect(HOST, PORT);

if (rc! = 0)
    logMessage ("rc from TCP connect is %d\r\n", rc);

MQTTPacket_connectData data = MQTTPacket_connectData_initializer;
```

```
data. MQTTVersion = 3;


// MQTT unique client id. Update this
data. clientID.cstring = "temp";
// MQTT user credentials. Update this
data. username. cstring = "vasilis1";
data. password. cstring = "vasilis1";


if ((rc = client. connect(data))! = 0)
   logMessage ("rc from MQTT connect is %d\r\n", rc);

MQTT: Message message;

char buf [100];
sprintf (buf, "%.1f", temp_sensor_value);

message.qos = MQTT: QOS0;
message. retained = false;
message.dup = false;
message. payload = (void*) buf;
message. payloadlen = strlen(buf)+1;

rc = client. publish (TEMPERATURE_TOPIC, message);

sprintf (buf, "%.1f", hum_sensor_value);
MQTT: Message message1;
```

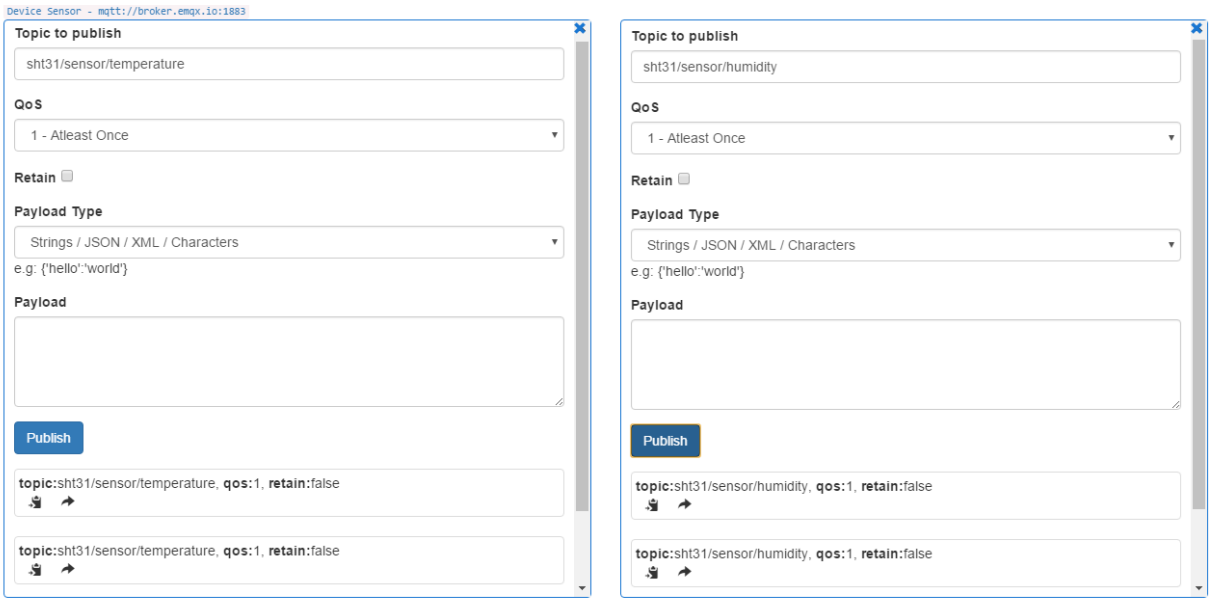
```
    message1.qos = MQTT: QOS0;
    message1.retained = false;
    message1.dup = false;
    message1.payload = (void*) buf;
    message1.payloadlen = strlen(buf)+1;

    rc = client. publish (HUMIDITY_TOPIC, message1);


    if ((rc = client. unsubscribe (TEMPERATURE_TOPIC)! = 0)
        logMessage ("rc from unsubscribe was %d\r\n", rc);

    temp = sht31.readTemperature();
        printf ("Temperature: %.2f C \n", temp);

    humidity = sht31.readHumidity();
        printf ("Humidity:  %.2f %% \n", humidity);

        led = temp > 25.0f;
        led1 = humidity > 40.0f;

    wait_ms (500);

    }
    return 0;
}
```

- Η δεύτερη θα έχει να κάνει με την λειτουργία subscribe(εγγραφή),όπου μέσα από το mqttbox θα κάνουμε publish και subscribe στον mbed simulator τις τιμές που έχουμε περάσει χειροκίνητα μέσα στον κώδικα για την θερμοκρασία και την υγρασία.

Το Interface του simulator για την λειτουργία subscribe



Το Interface του MqttBox



Main Code:

```
#define logMessage printf
#define MQTTCLIENT_QOS2 1
#include "Sht31.h"
#include "mbed.h"
#include "easy-connect.h"
#include "MQTTNetwork.h"
#include "MQTTmbed.h"
#include "MQTTClient.h"

int arrivedcount = 0;

float sample_sensor_value = 28.7;
float simple_sensor_value = 20.1;



void messageArrivedTemp (MQTT: MessageData& md)
{
   MQTT: Message &message = md. message;
   logMessage ("Message arrived: qos %d, retained %d, dup %d, packetid %d\r\n", message.qos, message. retained, message.dup, message.id);
   logMessage ("Temperature: %.1f C \r\n", sample_sensor_value);
   ++arrivedcount;

}

void messageArrivedHum (MQTT: MessageData& md)
{
   MQTT: Message &message = md. message;
```

```
    logMessage ("Message arrived: qos %d, retained %d, dup %d, packetid %d\r\n", message.qos,
message. retained, message.dup, message.id);
    logMessage ("Humidity: %.1f %% \r\n", simple_sensor_value);
    ++arrivedcount;

}


int main (int argc, char* argv [])
{
    // MQTT broker hostname
    const char* HOST = "broker.emqx.io";
    // MQTT broker port
    const int PORT = 1883;


    // MQTT topics for temperature & humidity
    const char* TEMPERATURE_TOPIC = "sht31/sensor/temperature";
    const char* HUMIDITY_TOPIC = "sht31/sensor/humidity";

    logMessage ("Hello MQTT Client");

    NetworkInterface* network = easy_connect(true);
    if (! network) {
        return -1;
    }

    MQTTNetwork mqttNetwork(network);
```

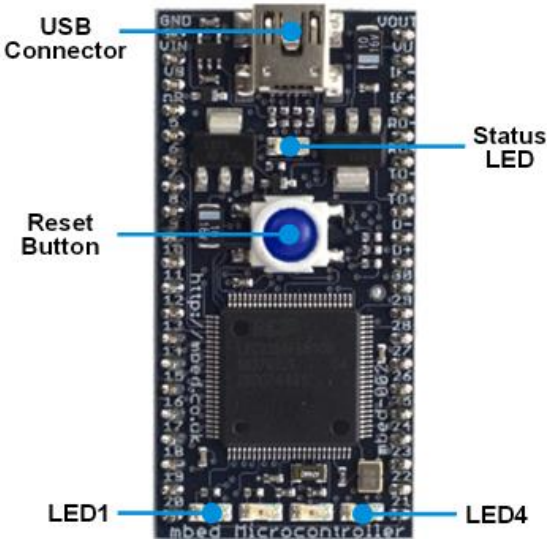
```
MQTT: Client<MQTTNetwork, Countdown> client(mqttNetwork);

logMessage ("Connecting to %s: %d\r\n", HOST, PORT);
int rc = mqttNetwork.connect(HOST, PORT);
if (rc! = 0)
   logMessage ("rc from TCP connect is %d\r\n", rc);

MQTTPacket_connectData data = MQTTPacket_connectData_initializer;
data. MQTTVersion = 3;

// MQTT unique client id. Update this
data. clientID.cstring = "temp";
// MQTT user credentials. Update this
data. username. cstring = "vasilis1";
data. password. cstring = "vasilis1";

if ((rc = client. connect(data))! = 0)
   logMessage ("rc from MQTT connect is %d\r\n", rc);

if ((rc = client. subscribe (TEMPERATURE_TOPIC, MQTT: QOS1, messageArrivedTemp)))
   if (rc! = 0)
   printf ("Failed to subscribe to MQTT topic. Return code: %d\n", rc);

if ((rc = client. subscribe (HUMIDITY_TOPIC, MQTT: QOS0, messageArrivedHum)! = 0)
   logMessage ("rc from MQTT subscribe is %d\r\n", rc);
   if (rc! = 0)
   printf ("Failed to subscribe to MQTT topic. Return code: %d\n", rc);
```

```
    while (true) {
    client. yield (1000); // Adjust the delay as needed
  }


  logMessage ("Arrived %d msgs\r\n", arrivedcount);

  return 0;
}
```

Κεφάλαιο 6 : Πως δουλεύει και πως εγκαθήστατε το Mbed Simulator

6.1 Εισαγωγή

Το Mbed είναι μια πλατφόρμα και ένα λειτουργικό σύστημα για συσκευές συνδεδεμένες στο διαδίκτυο που βασίζονται σε μικροελεγκτές ARM Cortex-M 32-bit. Τέτοιες συσκευές είναι επίσης γνωστές ως συσκευές Internet of Things. Το έργο αναπτύσσεται σε συνεργασία από την Arm και τους τεχνολογικούς εταίρους της. Υπάρχουν διάφορες πλακέτες επίδειξης υλικού για την πλατφόρμα Mbed, με την πρώτη να είναι η αρχική πλακέτα μικροελεκτή Mbed. Η πλακέτα μικροελεκτή Mbed (που διατίθεται στο εμπόριο ως "mbed NXP LPC1768") είναι μια πλακέτα επίδειξης που βασίζεται σε έναν μικροελεκτή της NXP, ο οποίος διαθέτει έναν πυρήνα ARM

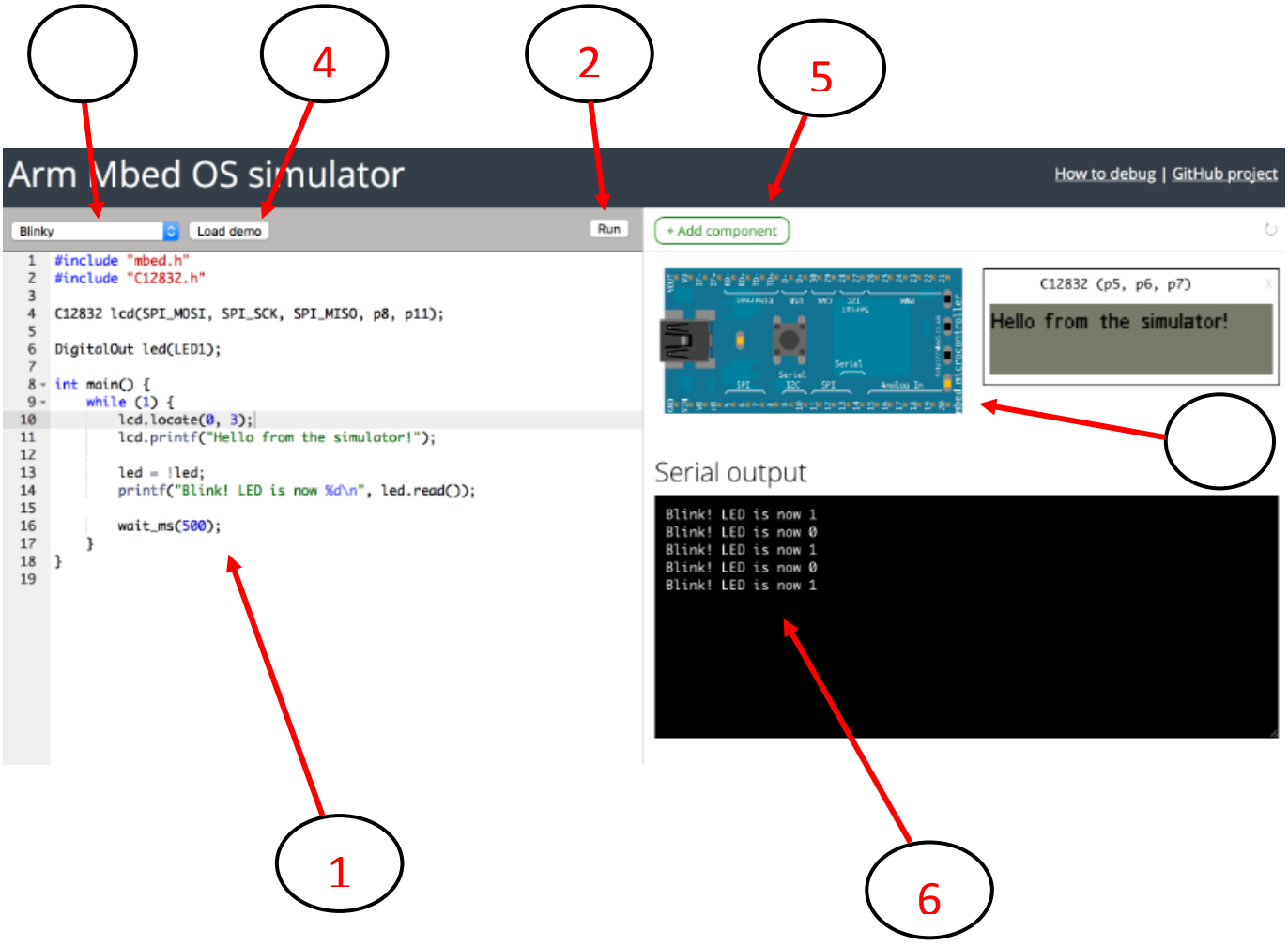


Cortex M3, που λειτουργεί στα 96 MHz, με 512 KB flash, 32 KB RAM, καθώς και διάφορες διεπαφές, όπως Ethernet, USB Device, CAN, SPI, I2C και άλλες I/O.

Εικόνα 3 : Πλακέτα NXP LPC1768

Ο προσομοιωτής σάς επιτρέπει να εκτελείτε εφαρμογές Mbed OS 5 απευθείας στον υπολογιστή σας, ώστε να μπορείτε να ελέγχετε και να επαληθεύετε γρήγορα τις εφαρμογές χωρίς να τις εκτελείτε σε πραγματική πλακέτα. Αυτό είναι ένα πολύτιμο εργαλείο εκμάθησης, καθώς μαθαίνετε γρήγορα πώς λειτουργεί το Mbed. Είναι επίσης πολύ χρήσιμο για την ανάπτυξη σύνθετων εφαρμογών. Στην Arm, έχουν χρησιμοποιήσει τον προσομοιωτή για εφαρμογές στο mbed-http, τη στοίβα Mbed LoRaWAN και το uTensor.

6.2 Περιβάλλον του Simulator



Περιγραφή

1: Ο προσομοιωτής δείχνει τον code editor στα αριστερά, οπού μπορούμε να αλλάξουμε τον κωδικό σε εκείνο το πεδίο.

2: Κάνουμε κλικ στο Run για να εκτελέσουμε τον τρέχον κώδικα στον προσομοιωτή.

3: Σε αυτό το πεδίο επιλέγουμε την εφαρμογή στο αναπτυσσόμενο μενού, όπου υπάρχει ένα ευρύ φάσμα διαθέσιμων demo-εφαρμογών, από περιφερειακές εφαρμογές (όπως η δημοφιλής οθόνη LCD C12832) έως εφαρμογές δικτύου.

4: Αφού επιλέξουμε την εφαρμογή στο αναπτυσσόμενο μενού, κάνουμε κλικ στο Φόρτωση(Load Demo) και η εφαρμογή φορτώνεται αυτόματα.

5: Στην επιλογή +Add component μπορούμε να προσθέσουμε εξάρτημα για την υλοποίηση του Demo .

6: Μέσω του Serial Output μπορούμε να δούμε τα αποτελέσματα που περνούμε από τον code editor

7: Εδώ έχουμε την αναπτυξιακή πλακέτα του Mbed πάνω στην οποία εκτελείτε ο κώδικας μας.

6.3 Πως δουλεύει

Σε αντίθεση με τα Γρήγορα Μοντέλα, ο Προσομοιωτής Mbed δεν προσομοιώνει πλήρως μια συσκευή Cortex-M. Αντ' αυτού, μεταγλωττίζει το Mbed OS 5 χρησιμοποιώντας Emscripten. Το Emscripten είναι ένας μεταγλωττιστής LLVM σε JavaScript, ο οποίος αναλαμβάνει ένα σύνθετο έργο C++ και παράγει κάτι που μπορεί να εκτελέσει σε ένα περιβάλλον διαδικτύου. Στη συνέχεια, όταν στον πίνακα πρέπει να κάνει μια εργασία (για παράδειγμα, να αποσυνδέσει ένα pin) η ενέργεια προσομοιώνεται και αντί να αλλάξουμε έναν καταχωρητή, μπορούμε να να αλλάξουμε ένα γραφικό στοιχείο στην οθόνη. Μια παρόμοια προσέγγιση που ακολουθούμε για τα επιμέρους τη στιγμή που πρέπει να εκτελεστεί η πραγματική ενέργεια, εκτελείται η πραγματική λειτουργία επικοινωνίας, μπλοκάρουμε την κλήση και στέλνουμε τα δεδομένα στο πρόγραμμα περιήγησης. Για παράδειγμα, με μια οθόνη LCD, διακόπτουμε όταν ο ρυθμιστής καρέ γεμίζει και οδηγούμε την εικόνα σε έναν πίνακα εικονοστοιχείων. Οτιδήποτε άλλο στο πρόγραμμα οδήγησης, από την απόδοση γραμματοσειρών μέχρι το API, παραμένει ακριβώς το ίδιο.

6.4 Εγκατάσταση

Ο προσομοιωτής διατίθεται σε δύο εκδοχές: μια online έκδοση, η οποία εκτελείται πλήρως στο πρόγραμμα περιήγησής σας, και μια offline έκδοση, η οποία λειτουργεί για οποιοδήποτε έργο

Mbed OS 5. Ο ευκολότερος τρόπος για να ξεκινήσετε είναι να ανοίξουμε τον online προσομοιωτή Mbed, όπου μας εμφανίζει το περιβάλλον που είδαμε παραπάνω.

Μπορούμε επίσης να εκτελέσουμε τον προσομοιωτή εκτός σύνδεσης σε οποιοδήποτε έργο Mbed OS 5. Αυτό μας επιτρέπει να ενσωματώσουμε τον προσομοιωτή στη ροή εργασιών ανάπτυξής μας. Για να χρησιμοποιήσουμε τον προσομοιωτή εκτός σύνδεσης, πρέπει να βεβαιωθούμε ότι έχουμε τα παρακάτω εργαλεία όπως:

1.Mbed CLI.

2.Git - βεβαιωθείτε ότι το git βρίσκεται στο PATH σας.

3.Mercurial - βεβαιωθείτε ότι το hg βρίσκεται στο PATH σας.

4.Το SDK Emscripten.

5.Node.js 8 ή νεότερη έκδοση.

Στη συνέχεια, ανοίγουμε ένα παράθυρο τερματικού, πλοηγούμαστε σε ένα έργο Mbed OS 5 και εκτελούμε τα παρακάτω:

installs the simulator

$ npm install mbed-simulator -g

run the current project

$ mbed-simulator.

Αυτό μεταγλωττίζει την τρέχουσα εφαρμογή και ανοίγει ένα πρόγραμμα περιήγησης που εκτελεί την εφαρμογή.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Τα ενσωματωμένα συστήματα αποτελούν αναπόσπαστο μέρος του σύγχρονου κόσμου. Είναι ειδικά κατασκευασμένα για συγκεκριμένες εφαρμογές, επιτρέποντας το σχεδιασμό και τη βελτιστοποίηση, καθιστώντας πιθανό για έναν πελάτη να επωφεληθεί από τη χρήση τους, ελαχιστοποιώντας παράλληλα την κατανάλωση ενέργειας και το κόστος. Μελέτες δείχνουν ότι τα ενσωματωμένα συστήματα έχουν επηρεάσει σε μεγάλο βαθμό την υγειονομική περίθαλψη, τον στρατό και τις επιχειρήσεις, ενώ τεχνολογίες και κινήματα όπως η βιομηχανική επανάσταση, το Διαδίκτυο των πραγμάτων και τα έξυπνα οχήματα και σπίτια συνεχίζουν να κερδίζουν έδαφος. Ο κλάδος των τηλεπικοινωνιών περιέχει τα σημαντικότερα μερίδια αγοράς εφαρμογών ενσωματωμένων συστημάτων που εφαρμόζονται σε δίκτυα που συνδέονται με ασύρματες επικοινωνίες και ένα τμήμα μικροελεκτών. Ορισμένες από τις τάσεις στα ενσωματωμένα συστήματα περιλαμβάνουν την τεχνητή νοημοσύνη, το Διαδίκτυο των πραγμάτων, την κυβερνοασφάλεια, τη μηχανική μάθηση, τις ταχύτητες και τη γλώσσα προγραμματισμού python. Οι τάσεις στα ενσωματωμένα συστήματα επηρεάζουν τις συστάσεις των πελατών που χρησιμοποιούν τα προϊόντα. Έχουν διαφορετικές ανάγκες και προτιμήσεις όπως η ασφάλεια και η εμπιστευτικότητα, η αποθήκευση δεδομένων, οι ταχύτητες απόδοσης, η κατανάλωση ενέργειας και το κόστος. Απεικονίζεται ότι τα ενσωματωμένα συστήματα εξελίσσονται, εστιάζοντας στην αύξηση των ταχυτήτων απόδοσης, στην ανάπτυξη συστημάτων κυβερνοασφάλειας, στη μικρότερη κατανάλωση ενέργειας και στην αποδοτικότητα κόστους. Επιπλέον, Το ενσωματωμένο σύστημα είναι ένας συνεχώς αναπτυσσόμενος τομέας, καθώς όλο και πιο ισχυροί μικροεπεξεργαστές γίνονται διαθέσιμοι σε χαμηλό κόστος, αναπτύσσονται όλο και πιο πολύπλοκα ενσωματωμένα συστήματα. Οι απαιτήσεις για τα ενσωματωμένα συστήματα περιλαμβάνουν μικρό μέγεθος, χαμηλό κόστος, λειτουργία σε πραγματικό χρόνο, ασφάλεια και αξιοπιστία. Ωστόσο, για να είναι επιτυχημένα τα μελλοντικά εργαλεία μπορεί να χρειαστεί να αυξήσουν ακόμη περισσότερο το πεδίο εφαρμογής, ώστε να συμπεριλάβουν θέματα κύκλου ζωής και επιχειρηματικά θέματα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Wayne Wolf, “Computers as Components: Principles of Embedded Computer Systems”, Elsevier, 2005, Ελληνική Μετάφραση: “Οι υπολογιστές ως συστατικά στοιχεία: Βασικές Αρχές Σχεδίασης Ενσωματωμένων Συστημάτων”, Εκδόσεις Νέων Τεχνολογιών, 2007.

[2] William Stallings, Λειτουργικά συστήματα, Αρχές σχεδίασης, Εκδότης: Τζιόλα,2017.

[3] Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel, O'Reilly Media, Inc,2005.

[4] Δασυγένης, Μ., & Σουντρής, Δ. (2015), Ενσωματωμένα συστήματα [Undergraduate textbook]. Kallipos, Open Academic Editions. https://hdl.handle.net/11419/2247.

[5] https://www.watelectronics.com/arm-processor-architecture-working/

[6] https://www.redhat.com/en/topics/linux/what-is-arm-processor

[7]https://www.zdnet.com/article/arm-processors-everything-you-need-to-know-now/

[8]https://www.bluefruit.co.uk/training/a-brief-history-of-embedded-operating-systems/

[9]https://www.ultralibrarian.com/2022/06/28/types-of-embedded-systems-characteristics-classifications-ulc

[10]https://electricalfundablog.com/embedded-system-characteristics-types-advantages-disadvantages/

[11]https://www.infopulse.com/blog/challenges-and-issues-of-embedded-software-development

[12] https://www.digi.com/blog/post/examples-of-embedded-systems

[13] https://www.totalphase.com/blog/2022/03/what-is-arm-processor-comparison-x86-and-advantages-disadvantages/

[14] https://www.windriver.com/solutions/learning/embedded-linux

[15] https://blog.felgo.com/embedded/embedded-operating-systems