

**Κατασκευή συστημάτων επικοινωνίας
μεγάλης εμβέλειας και χαμηλής
κατανάλωσης με τη χρήση nucleo**

Γρηγοριάδης Νικόλαος-Χρυσοβαλάντης
Αριθμός Μητρώου: 3208

Διπλωματική Εργασία

Επιβλέπων: Χρυσοβαλάντης Καβουσιανός

Ιωάννινα, Σεπτέμβριος, 2022



ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

Ευχαριστίες

Η παρούσα διπλωματική εργασία σηματοδοτεί την ολοκλήρωση του προπτυχιακού προγράμματος σπουδών μου στο τμήμα << Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής, Πολυτεχνική Σχολή – Πανεπιστημίου Ιωαννίνων >>. Η ολοκλήρωση της διπλωματικής εργασίας ωστόσο, δεν θα ήταν δυνατή χωρίς την πολύτιμη στήριξη του καθηγητή μου, Κυρίου Χρυσοβαλάντη Καβουσιανού, όπου οφείλω ένα μεγάλο ευχαριστώ για όλη την βοήθεια που μου πρόσφερε, αλλά και για το γεγονός ότι μέσω των μαθημάτων του μου κίνησε το ενδιαφέρον να διαβάσω και να γνωρίσω περισσότερο τον όμορφο κόσμο του hardware. Ευχαριστώ πολύ βαθιά την σύντροφό μου Έλενα Θεοδώρου(Διπλωματούχος Φυσικοθεραπείας, στο Πανεπιστήμιο Δυτικής Αττικής) για την συνεχή συμπαράσταση, την έμπνευση αλλά και την κατανόηση που έδειξε κοντά μου σε όλη αυτή την διάρκεια. Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου, Κώστα, Κατερίνα και Δήμητρα οι οποίοι υπήρξαν ένα ανεκτίμητο στήριγμα για εμένα και στους οποίους οφείλω όλη την διαδρομή των σπουδών μου, μέχρι σήμερα.

Ιωάννινα, Σεπτέμβριος 2022

Γρηγοριάδης Νικόλαος Χρυσοβαλάντης.

Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται την ανάπτυξη ενός συστήματος που έχει ως κύρια πηγή της δύο μικροελεγκτές της εταιρίας ST Microelectronics, STM32L053R8. Κύριος στόχος είναι τόσο η επικοινωνία των δύο μικροελεγκτών (STM32), όσο και η εξοικονόμηση ενέργειας στο κάθε σύστημα, το οποίο απαρτίζεται από μία οθόνη LCD, έναν ασύρματο πομποδέκτη nRF24L01+ των 2.4 GHz, ένα πληκτρολόγιο και ένα μικρόφωνο. Η ανάλυση ξεκινά με τα κύρια χαρακτηριστικά του μικροελεγκτή όπου αποτελεί την βάση για την σύνδεση και την επεξεργασία δεδομένων με τον ασύρματο πομποδέκτη. Οι υποδείξεις της κατασκευαστικής εταιρίας για τις καταστάσεις χαμηλής ενέργειας θα αποτελέσουν βασικό πυλώνα στην επίτευξη του στόχου, καθώς επίσης και άλλα χαρακτηριστικά του όπως η δομή του πυρήνα Corter-M0+ που επεξηγεί τις προδιαγραφές του μικροελεγκτή για μια πιο ολοκληρωμένη κατανόηση. Έπειτα, γίνεται αναφορά στις λειτουργίες του ασύρματου πομποδέκτη, που είναι η εκτέλεση του πρωτεύοντος δέκτη (PX_MODE) και του πρωτεύοντος αποστολέα (RX_MODE). Κύρια έμφαση δίνεται στην εφαρμογή του πρωτοκόλλου Enhanced Shockburst, αλλά και τον τρόπο εναλλαγής καταστάσεων αποστολής (PX mode) και λήψης (RX mode) που είναι αρκετά σημαντικός για την σωστή επικοινωνία των δύο συσκευών. Τέλος, παρουσιάζεται ένα διάγραμμα από το σύστημα των δύο μικροελεγκτών με έμφαση στην υλοποίηση του.

Λέξεις Κλειδιά: ST Microelectronics, STM32L053R8, STM32, nRF24L01+, μικροελεγκτές, πρωτεύον δέκτης, πρωτεύων αποστολέας, πυρήνας M0+.

Abstract

This thesis deals with the development of a system that has as its main source two micro-controllers of the company ST Microelectronics, STM32L053R8. The main goal is, the communication of the two micro-controllers (STM32), as well as the energy saving in each system consisting of an LCD screen, a wireless nRF24L01+ 2.4 GHz transceiver, a keyboard and a microphone. The analysis begins with the main features of the micro-controller that forms the basis for connecting and processing data with the wireless transceiver. The manufacturer's hints for low-power states will be a key pillar in achieving the goal, as well as other features such as the Cortex-M0+ core structure that explains the micro-controller's specifications for a more comprehensive understanding. There is also a reference to the functions of the wireless transceiver, which are the execution of the primary receiver (RX_MODE) and the primary transmitter (TX_MODE). Main emphasis is given on the implementation of the Enhanced Shockburst protocol and the way of switching between sending (TX mode) and receiving states (RX mode) that are quite important for the correct communication between the two devices. Finally, it is presented a diagram of the system of the two micro-controllers with an emphasis on its implementation.

Keywords: ST Microelectronics, STM32L053R8, STM32, nRF24L01+, micro-controllers, primary receiver (RX_MODE), primary transmitter (TX_MODE), Cortex-M0+.

Υ

Πίνακας Περιεχομένων

Κεφάλαιο 1. Εισαγωγή	9
Κεφάλαιο 2. Μικροελεγκτής STM32.....	11
2.1 Η Αρχιτεκτονική ARM	12
2.1.1 Το μοντέλο προγραμματισμού της ARM.....	13
2.2 Ο πυρήνας Cortex-M0+	14
2.2.1 Η Δομή του Συστήματος	16
2.2.2 Οργάνωση Μνήμης.....	17
2.2.3 Εξαιρέσεις και διακοπές συστήματος	20
2.2.4 Ελεγκτής διακοπών NVIC	21
2.2.5 Οι ταλαντωτές του συστήματος.....	23
2.2.6 Λειτουργίες χαμηλής κατανάλωσης	25
2.3 Περιφερειακά Γενικού Σκοπού.....	28
2.3.1 Είσοδοι/Έξοδοι γενικού σκοπού GPIO's.....	28
2.3.2 Μονάδα εξωτερικών διακοπών συστήματος EXTI.....	29
2.3.3 Χρονιστές/Μετρητές Γενικού Σκοπού	30

2.4 Περιφερειακά επικοινωνίας	31
2.4.1 Πρωτόκολλο επικοινωνίας SPI.....	31
2.4.2 Πρωτόκολλο επικοινωνίας I2C	34
2.4.3 Πρωτόκολλο επικοινωνίας USB.....	37
2.4.4 Σειριακή επικοινωνία USART.....	38
Κεφάλαιο 3. nRF24L01+ ο Ασύρματος Πομποδέκτης	40
3.1 Ραδιοέλεγχος.....	41
3.1.1 Power Down - κατάσταση απενεργοποίησης.....	43
3.1.2 Standby - κατάσταση αναμονή	43
3.1.3 Receiver - κατάσταση λήψης.....	43
3.1.4 Transmitter - κατάσταση μετάδοσης.....	44
3.2 Έλεγχος δεδομένων και περιφερειακού.....	44
3.2.1 Εντολές διεπαφής SPI	45
3.3 Ουρές δεδομένων	46
3.4 Φυσικό κανάλι.....	47
3.4.1 Ρυθμός μετάδοσης δεδομένων στον αέρα	47
3.4.2 Ραδιοσυχνότητα καναλιού RF.....	48
3.4.3 Ανιχνευτής ισχύος RPD	48
3.5 Πρωτόκολλο Enhanced Shockburst	49
3.4.1 Πακέτο μετάδοσης πρωτοκόλλου	50
3.4.2 Αυτόματες συναλλαγές πακέτων.....	52
3.4.3 Η λειτουργία του πρωτεύων αποστολέα (PTX).....	52
3.4.4 Η λειτουργία του πρωτεύων δέκτη (PRX)	53
3.4.5 Η λειτουργία πολλαπλής λήψης (multireceiver).....	54
Κεφάλαιο 4. Υλοποίηση	56
4.1 Η επικοινωνία του μικροελεγκτή με τον ασύρματο πομποδέκτη - SPI.....	57
4.1.1 Η ενεργοποίηση της υπορουτίνας ISR.....	58
4.2 Η εφαρμογή του πρωτοκόλλου Enhanced Shockburst.....	59
4.2.1 Ενεργοποίηση των καταστάσεων TX και RX.....	60

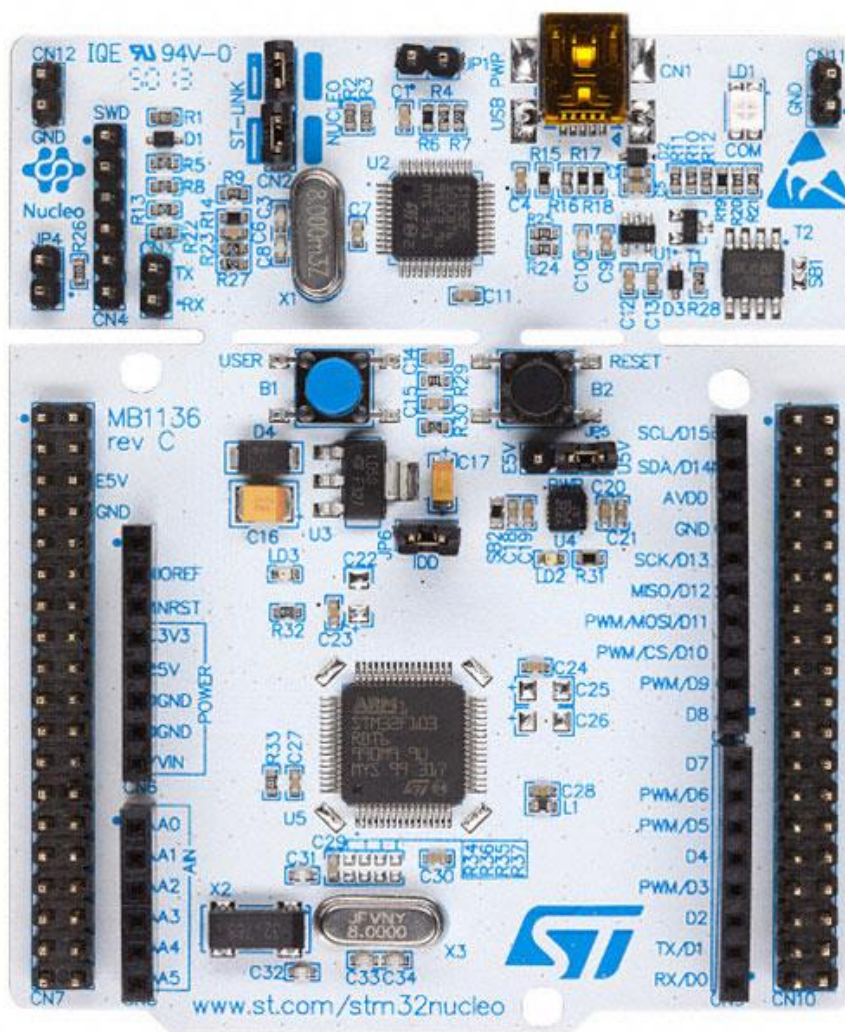
4.2.2 Αρχικοποίηση του πρωτοκόλλου.....	61
4.2.3 Ανάγνωση ωφέλιμου φορτίου	61
4.3 Switch debouncing.....	62
4.3.1 Πληκτρολόγιο.....	64
4.4 Επικοινωνία μικροελεγκτή με οθόνη LCD - I2C.....	67
4.5 Μείωση κατανάλωσης – Sleep Mode	70
4.6 Ρολόι συστήματος.....	71
Κεφάλαιο 5. Συζήτηση – Συμπεράσματα.....	73
Πείραμα 1 Μείωση των παλμών ρολογιού	73
Πείραμα 2 Ενεργοποίηση low power κατάστασης και απενεργοποίηση περιφερειακών	75
Συμπέρασμα.....	76
Παράρτημα Α Παραδείγματα κώδικα.....	78
A.1 Ενεργοποίηση ρολογιού PLL ως ρολόι συστήματος-RCC.	78
A.2 Low power – Sleep mode.....	79
A.2.1 Μείωση παλμών ρολογιού – Ενεργοποίηση MSI clock	79
A.2.2 Απενεργοποίηση LED οθόνης.....	80
A.3 Αρχικοποίηση SPI.....	80
A.3.1 SPI λήψη και αποστολή δεδομένων με interrupt.....	81
A.3.2 Εξασφάλιση κατάστασης RX.....	82
A.3.3 Διάβασμα ωφέλιμου φορτίου	82
A.4 Αρχικοποίηση I2C	83
A.5 Αντιμετώπιση Switch debouncing.....	84
A.6 Αρχικοποίηση Enhanced Shockburst.....	84
A.7 Προγραμματισμός εξωτερικού σήματος διακοπής.....	86

όπως για παράδειγμα οι μικροελεγκτές. Τόσο η αύξηση της πυκνότητας, όσο και οι υψηλότερες ταχύτητες λειτουργίας οδηγούν στον σχεδιασμό περίπλοκων ολοκληρωμένων κυκλωμάτων (IC – Integrated Circuit) με υψηλές συχνότητες ρολογιού, πράγμα που αυξάνει επίσης την κατανάλωση ενέργειας στο σύστημα και ταυτόχρονα επιτυγχάνεται και η αύξηση της θερμοκρασίας. Έτσι, η θερμότητα που διαχέεται πρέπει να αφαιρεθεί, ώστε να διατηρηθεί η θερμοκρασία του μικροελεγκτή σε αποδοτικά επίπεδα, οπότε το κόστος της ψύξης και της απομάκρυνσης αποτελούν σημαντικό παράγοντα σε αυτή την προσπάθεια. Πολλοί μικροεπεξεργαστές που σχεδιάστηκαν την δεκαετία του 1990, λειτουργούν με συχνότητες που ποικίλουν ανάμεσα στις τιμές των 100 έως 300 MHz με μια τυπική κατανάλωση ενέργειας να είναι της τάξης των 20 με 50 W.

Κεφάλαιο 2. Μικροελεγκτής STM32

Η παρούσα διπλωματική εργασία είναι βασισμένη σε ένα προϊόν της εταιρείας ST Microelectronics που αφορά το development board nucleo-STM32, το οποίο φιλοξενεί τον μικροελεγκτή STM32L053R8. Εκείνος με την σειρά του κατατάσσεται στην οικογένεια μικροελεγκτών STM32L053x6/8 με εξαιρετικές επιδόσεις χαμηλής κατανάλωσης ισχύος που προσφέρει ο ενσωματωμένος Arm® Cortex®-M0+ 32-bit RISC πυρήνας που λειτουργεί σε συχνότητα 32MHz. Διαθέτει μνήμες με αξιοσημείωτα υψηλές ταχύτητες (φτάνουν μέχρι τα 64Kbytes Flash μνήμης, 2Kbyte EEPROM και 8 Kbyte που αφορούν την μνήμη RAM), έχοντας επιπλέον και τις νέες πιο βελτιωμένες πηγές εισόδων/εξόδων και περιφερειακών που συνδέονται μέσω τριών διαύλων Advanced Peripheral Bus ή αλλιώς APB που ονομάζονται APB1, APB2 και AHB. Οι μικροελεγκτές της συγκεκριμένης οικογένειας προσφέρουν υψηλές αποδόσεις στην ισχύ τους με την βοήθεια τόσο των εσωτερικών-εξωτερικών πηγών ρολογιού όσο και με την προσαρμογή τάσης σε αρκετές λειτουργίες που αφορούν την μείωση της κατανάλωσης στην εκάστοτε εφαρμογή που αναπτύσσεται στην συγκεκριμένη πλακέτα. Εμβαθύνοντας, διατίθενται προς χρήση ένα ADC 12bit, ένα DAC, δύο συγκριτές με χαμηλές επιδόσεις, καθώς επίσης έναν χρονοδιακόπτη χαμηλής κατανάλωσης με την ονομασία LPTIM, τρεις χρονοδιακόπτες γενικού σκοπού και έναν SysTick χρονοδιακόπτη. Όσον αφορά τις διεπαφές επικοινωνίας διαθέτει δύο I2C (Inter-Integrated Circuit), δύο SPIs (Serial Peripheral Interface), ένα I2S (Inter-IC Sound), δύο USART (Universal Synchronous/Asynchronous Receiver/Transmitter), ένα UART (Universal Asynchronous Receiver Transmitter) και ένα USB (Universal Serial Bus). Οι συσκευές που χρησιμοποιούν τους συγκεκριμένους μικροελεγκτές λειτουργούν από 1,8V έως 3,6V με την επιλογή BOR (Brown Out Reset) και από 1,65V έως 3,6V χωρίς την επιλογή BOR. Αξίζει να σημειωθεί ότι αρκετοί μικροελεγκτές διαθέτουν στο κύκλωμά τους μια προστασία, όπου όταν η τάση που τροφοδοτεί το κύκλωμα πέσει, τότε οδηγεί το σύστημα σε μια κατάσταση που ονομάζεται reset με σκοπό να διασφαλίσει την σωστή εκκίνηση όταν το ρεύμα επιστρέψει. Αυτή η ασφάλεια

ονομάζεται Brown Out Reset ή αλλιώς BOR. Τέλος, οι επιτρεπτές θερμοκρασίες λειτουργίας του συγκεκριμένου development board είναι μεταξύ των -40 έως 125 °C, όπου ένα σύνολο λειτουργιών εξοικονόμησης ενέργειας επιτρέπει τον σχεδιασμό εφαρμογών χαμηλής κατανάλωσης. Αυτός, άλλωστε, είναι και ο σκοπός της συγκεκριμένης παρουσίασης.



Εικόνα 1 Development Board Nucleo STM32L053R8

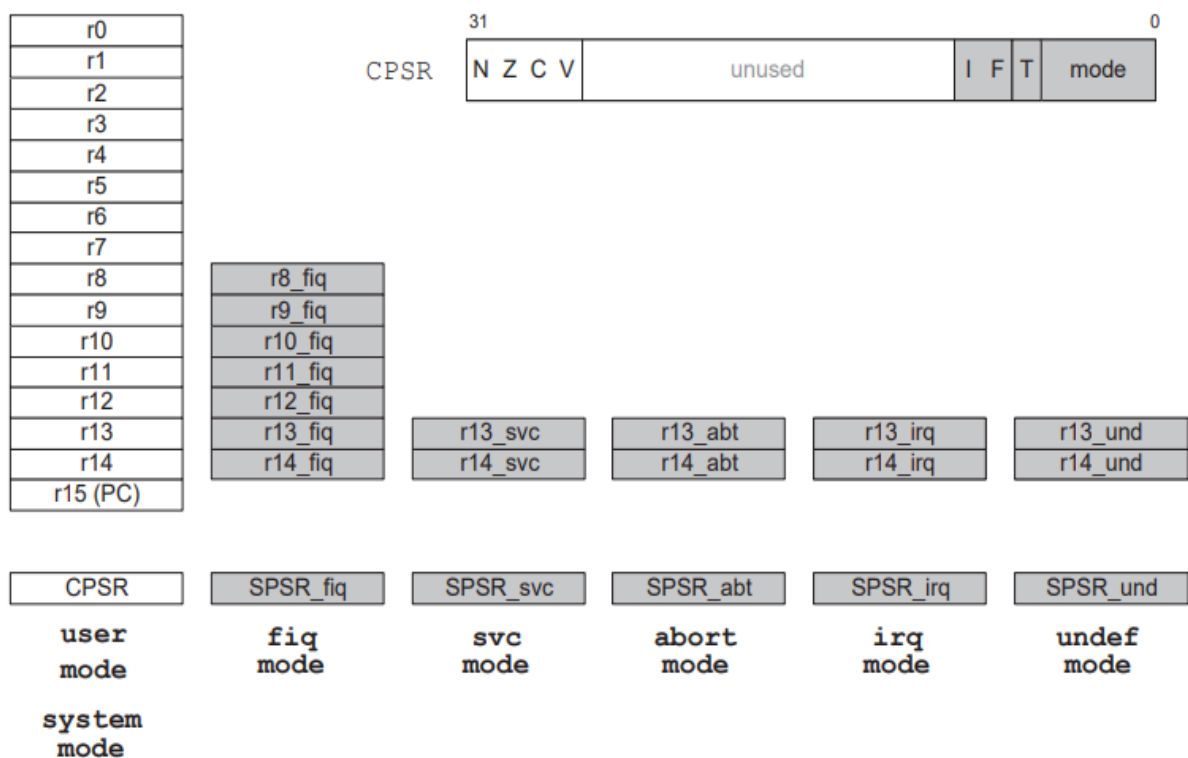
2.1 Η Αρχιτεκτονική ARM

Η ARM (Advanced Risc Machine) αρχιτεκτονική είναι ένας Reduce Instruction Set Computer ή αλλιώς RISC επεξεργαστής, του οποίου η ιστορία ξεκινά από την εταιρεία Acorn Computers το έτος 1983. Σύμφωνα με τον συγγραφέα Markus Levy η εταιρεία είχε ανάγκη από έναν 16-bit μικροελεκτή ο οποίος θα κάλυπτε όλες τις πολύπλοκες ανάγκες εκείνης της εποχής, που χρειάζονταν εκατοντάδες κύκλους ρολογιού για την εκτέλεσή τους. Με τα δεδομένα εκείνα οι επεξεργαστές που είχανε

στην κατοχή τους ήτανε τόσο αργοί που οδηγούσανε την εκτέλεση εντολών σε υψηλές καθυστερήσεις. Έτσι, λοιπόν, οι μηχανικοί της Acorn σχεδίασαν τον δικό τους επεξεργαστή που θα επέτρεπε την δημιουργία μικρών αλλά και υψηλά δυναμικών σε απόδοση υλοποιήσεων. Η βασική του δομή ξεκινά με κάποια κύρια χαρακτηριστικά του RISC επεξεργαστή όπως είναι η αρχιτεκτονική load/store που επιτρέπει την διαχείριση των δεδομένων να γίνεται μεταξύ των καταχωρητών και όχι απευθείας με την κύρια μνήμη, ένα σταθερό σε μήκος πεδίο εντολών στα 32-bit και τρεις μορφές διευθύνσεων στις εντολές. Πάνω σε αυτά τα χαρακτηριστικά ενσωματώθηκαν κάποιες νέες τεχνολογίες που καθιστούν την ARM αρχιτεκτονική μια από τις πιο αξιόπιστες λύσεις για απαιτητικές, χαμηλού κόστους και υψηλών αποδόσεων υλοποιήσεις. Ορισμένα από αυτά είναι ο έλεγχος της ALU ή αλλιώς Arithmetic Logic Unit και του καταχωρητή μετατόπισης δεδομένων, σε κάθε εντολή επεξεργασίας για την μεγιστοποίηση της απόδοσής τους. Ακόμη, αναπτύχθηκε περαιτέρω η αρχιτεκτονική load/store προκειμένου να γίνονται πολλές και ταυτόχρονες ενέργειες αναζήτησης και εγγραφής, ενώ ακόμη τοποθετήθηκε και αυτόματο σύστημα αύξησης/μείωσης στους τρόπους λειτουργίας των διευθύνσεων.

2.1.1 Το μοντέλο προγραμματισμού της ARM

Οι εντολές ενός επεξεργαστή ορίζουν λειτουργίες που ένας προγραμματιστής μπορεί να χρησιμοποιήσει ώστε να διαφοροποιήσει την κατάσταση του συστήματος μέσα στον επεξεργαστή. Η ARM διαθέτει δεκαέξι καταχωρητές γενικού σκοπού στον προγραμματισμό επιπέδου χρήστη. Κάποιοι από αυτούς χαρακτηρίζονται από επιπλέον ικανότητες, όπως αρχικά ο καταχωρητής r13 που θεωρητικά λειτουργεί και ως δείκτης στοίβας (stack pointer). Έπειτα, ο r15 είναι ο program counter και ο καταχωρητής CPSR (Current Program Status Register), ο οποίος διαθέτει κάποιες σημαίες (flags) τύπου συνθήκης (condition) όπως ('Negative','Zero','Carry','Overflow') όπου αντικατοπτρίζουν την κατάσταση εκτέλεσης του επεξεργαστή. Αξίζει να σημειωθεί ότι στο επίπεδο χρήστη μόνο οι πρώτοι δεκαπέντε καταχωρητές καταχωρούνται, είναι ορατοί και πρέπει να ληφθούν υπόψη. Οι υπόλοιποι καταχωρητές χρησιμοποιούνται μόνο για την επεξεργασία προγραμμάτων σε επίπεδο συστήματος και είναι υπεύθυνοι για την διαχείριση των εξαιρέσεων όντας και μη ορατοί προς τον χρήστη.

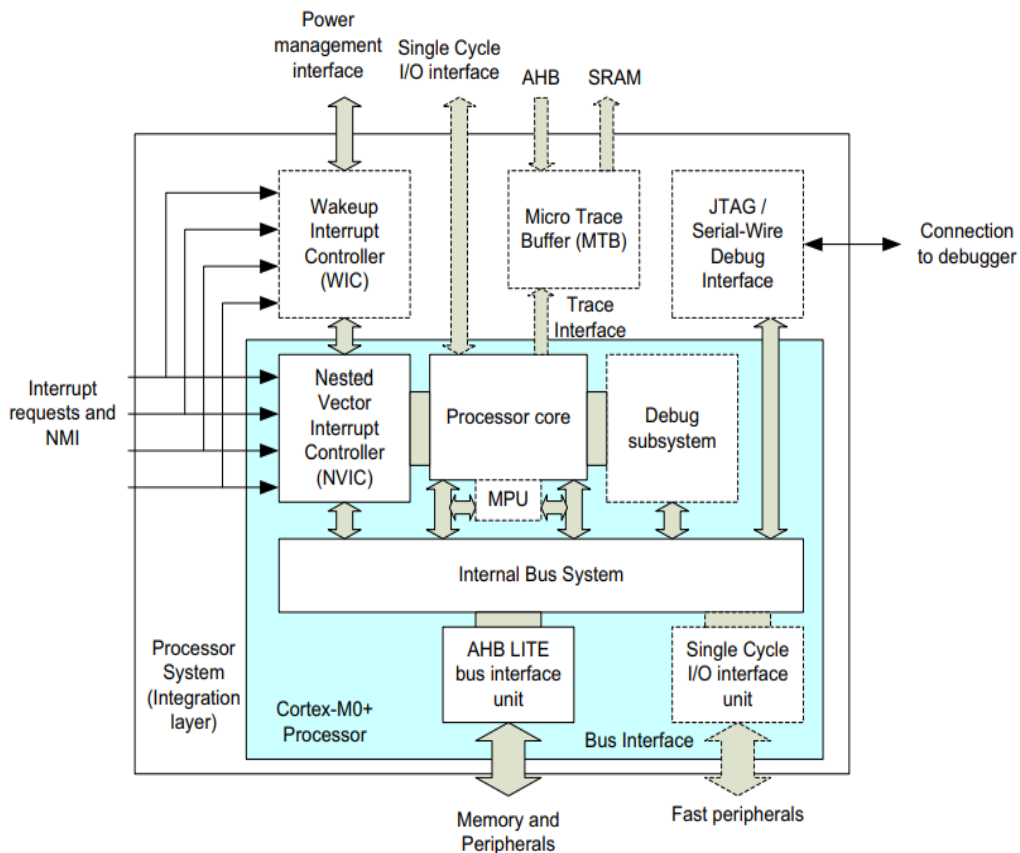


Εικόνα 2 Οι ARM καταχωρητές

2.2 Ο πυρήνας Cortex-M0+

Ο Cortex-M0+ είναι ο πιο ενεργειακά αποδοτικός επεξεργαστής στον σχεδιασμό συστημάτων που έχουν στόχο την ανάπτυξη εφαρμογών χαμηλού κόστους και μειωμένης κατανάλωσης. Τα 32-bit του επεξεργαστή προσφέρουν αρκετά πλεονεκτήματα σε έναν προγραμματιστή και καθιστούν την αρχιτεκτονική του απλή τόσο στην κατανόηση όσο και στον προγραμματισμό της. Έχει μια σύνδεση εισόδου/εξόδου σε λειτουργία ενός κύκλου που θέτει την πρόσβαση στα περιφερειακά από τους καταχωρητές γρηγορότερη. Έτσι, λοιπόν, η ανάπτυξη των εφαρμογών είναι βασισμένη σε έναν υψηλής απόδοσης πυρήνα με αρχιτεκτονική σωλήνωσης (pipeline) limited Harvard. Η σωλήνωση του συγκεκριμένου επεξεργαστή αποτελείται από δύο στάδια (fetch + predecode, decode+execute) που ονομάζεται von Neumann. Υλοποιεί την αρχιτεκτονική ARMv6-M που έχει βάση το σύνολο εντολών Thumb 16-bit, περιλαμβάνοντας τεχνολογία Thumb-2. Το γεγονός αυτό παρέχει μια αξιόπιστη και αποδοτική λύση μιας 32-bit αρχιτεκτονικής που κατέχει μεγαλύτερη πυκνότητα κώδικα

σε σχέση με τους μικροελεγκτές 8-bit και 16-bit. Επιπλέον, ενσωματώνει στον πυρήνα του το Nested Vector Interrupt Controller (NVIC), όπου προσφέρει κορυφαία απόδοση σε διακοπές (interrupts) αλλά και σε εξαιρέσεις (exceptions). Διαθέτει τριάντα δύο τύπους διακοπών (interrupt) από πολλά ήδη περιφερειακών, παρέχοντας την δυνατότητα της επιλογής προτεραιότητας για την διαχείριση τους, την επιλογή χρήσης διακοπής χωρίς μάσκα Non-Maskable Interrupt (NMI) αλλά και αρκετά σήματα διακοπής του ίδιου του συστήματος που χαρακτηρίζονται ως εξαιρέσεις. Στο λειτουργικό σύστημα περιλαμβάνεται ο χρονοδιακόπτης SysTick (System Tick Timer) για την λειτουργία της περιοδικής διατήρησης χρόνου. Ο επεξεργαστής υποστηρίζει προνομιακό και μη προνομιακό επίπεδο εκτέλεσης. Το γεγονός αυτό επιτρέπει στο λειτουργικό σύστημα να εκτελεί ορισμένες από τις εργασίες της εφαρμογής σε μη προνομιακό επίπεδο, πράγμα που θέτει περιορισμούς στην εκάστοτε εφαρμογή π.χ. σε θέματα πρόσβασης στην μνήμη. Συνεχίζοντας, ένα ακόμη κύριο χαρακτηριστικό είναι τα δύο ήδη χαμηλής κατανάλωσης που προσφέρονται όπως είναι το normal sleep και το deep sleep όπου μπορεί κανείς να τα ενεργοποιήσει με τις εντολές WFI (Wait For Interrupt) και WFE (Wait For Event). Μια πρόσθετη λειτουργία είναι ο ελεγκτής αφύπνισης με χρήση interrupt WIC. Το σύστημα για την αποσφαλμάτωση βασίζεται στο ARM CoreSight Debug Architecture που προσφέρεται για πολύπλοκα πολυπύρηνια σχεδιαστικά συστήματα επεξεργαστών. Η χρήση του μπορεί να γίνει μέσω του πρωτοκόλλου JTAG με τέσσερα ή πέντε pin ή σειριακά με δύο pin. Προσφέρει επίσης την δυνατότητα να χρησιμοποιήσουμε τέσσερα hardware breakpoints, δύο data breakpoints και απεριόριστα breakpoints στο προγραμματιστικό κομμάτι. Τέλος, το σύστημα επικοινωνίας βασίζεται στην μορφή διαύλων και αποτελείται από το δίαυλο AHB-Lite και από το internal bus system, το μέγεθος των οποίων είναι 32-bit και είναι δημιουργία της ARM.

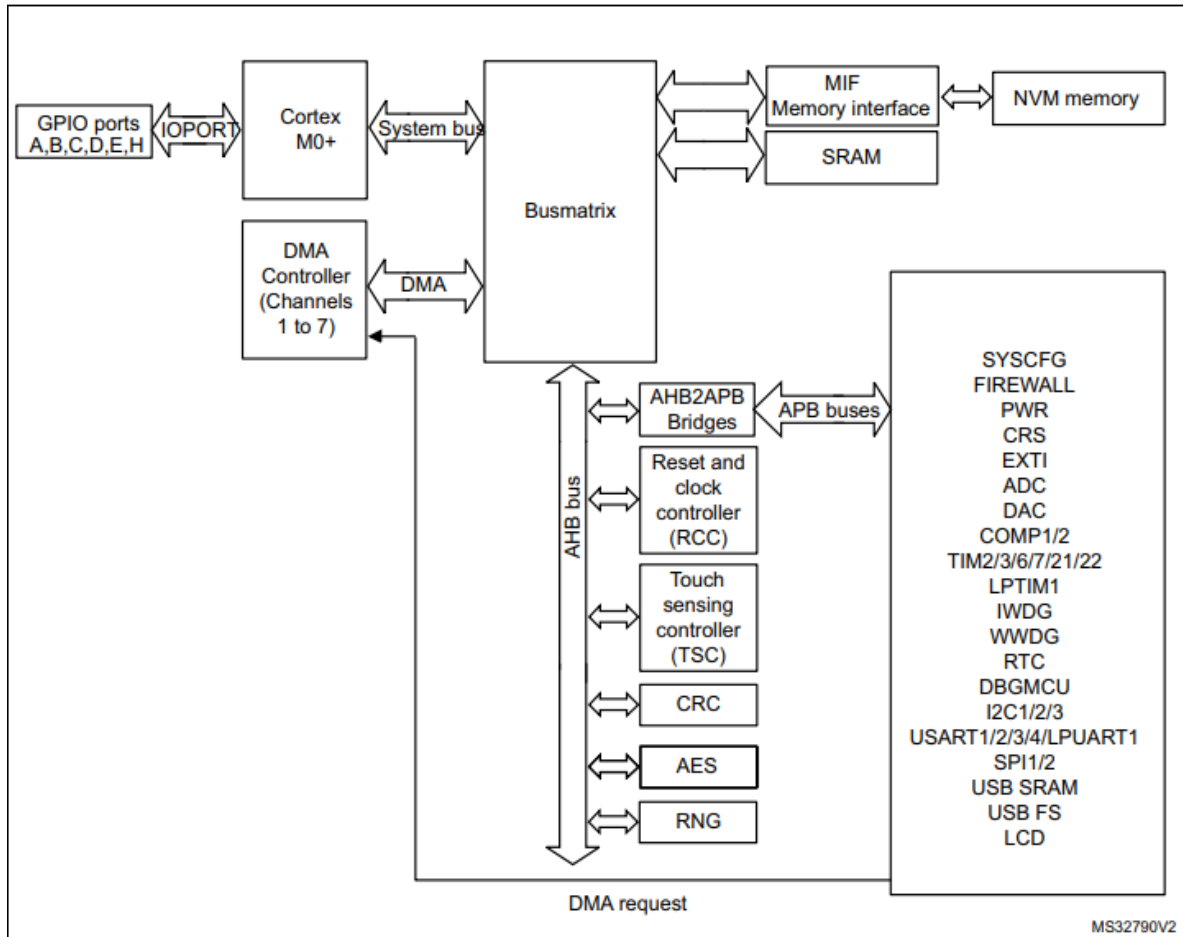


Εικόνα 3 Ο πυρήνας Corter-M0+

2.2.1 Η Δομή του Συστήματος

Παρατηρώντας την αρχιτεκτονική του συστήματος και σύμφωνα με το datasheet του μικροελεγκτή βλέπουμε ότι αποτελείται από δύο κατηγορίες που χαρακτηρίζονται ως αφέντης [master (Cortex-m0+, DMA)] και σκλάβος [slave (Internal SRAM, NVM memory, AHB-APB buses)]. Όσον αφορά εκείνης του master ο cortex-m0+ επεξεργαστής και η μονάδα DMA είναι άμεσα συνδεδεμένοι με τον δίαυλο Busmatrix, με την βοήθεια δύο διαύλων. Ο πρώτος συνδέει τον επεξεργαστή δίαυλος συστήματος (System Bus) και ο δεύτερος συνδέει την μονάδα DMA με το δίκτυο διαύλων. Στην ουσία, το BusMatrix είναι ο συνδετικός κρίκος μεταξύ των masters και των slaves, επιτρέποντας την ταυτόχρονη προσπέλαση και την αποτελεσματική λειτουργία ακόμη και όταν αρκετά περιφερειακά υψηλής ταχύτητας λειτουργούν ταυτόχρονα. Είναι σημαντικό να αναφερθεί ότι ο συγκεκριμένος δίαυλος βασίζεται στον αλγόριθμο Round Robin, του οποίου οι διεργασίες συστήματος παίρνουν χρόνο στον επεξεργαστή για να δώσουν τους πόρους τους σύμφωνα με ένα συγκεκριμένο κβάντο χρόνου. Από την

άλλη, οι δίαυλοι AHB και APB, με τον δεύτερο να απαρτίζεται από δύο υποκατηγορίες τον APB1 και τον APB2 που λειτουργούν με μέγιστη συχνότητα τα 32MHz, είναι το κλειδί για την επικοινωνία των περιφερειακών.



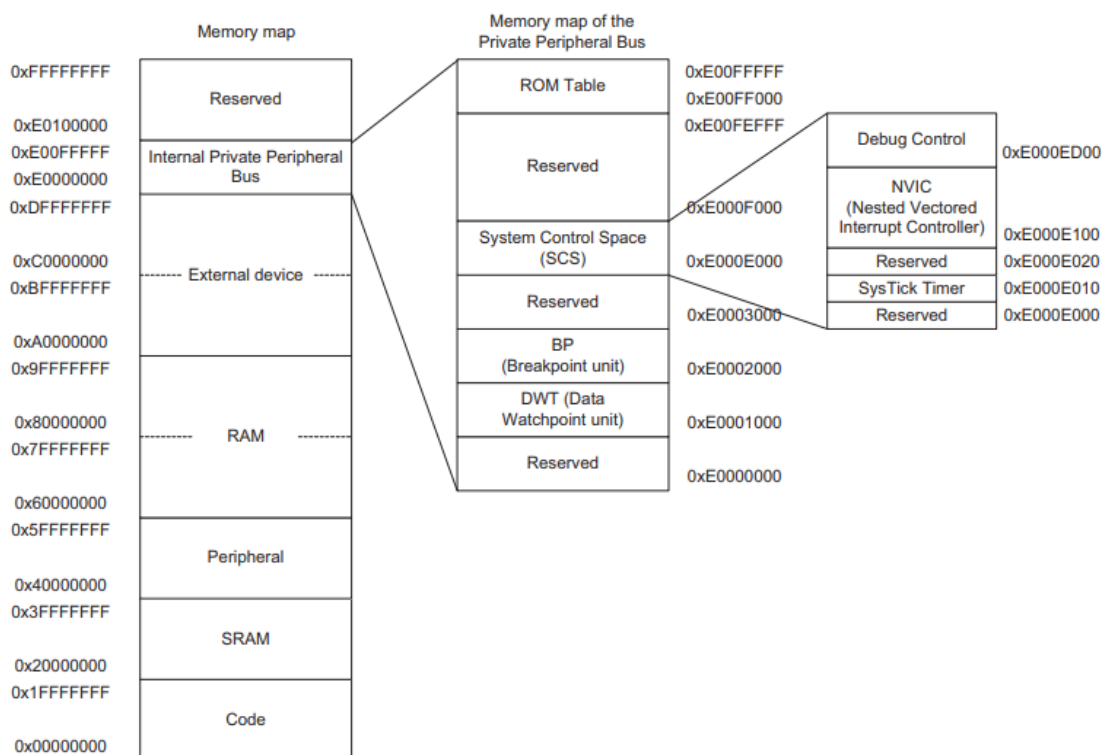
Εικόνα 4 Αρχιτεκτονική διαύλων

2.2.2 Οργάνωση Μνήμης

Ο Cortex-M0+ επεξεργαστής παρέχει 4GB προσπελάσιμης μνήμης, που διαχωρίζονται σε περιοχές, δημιουργώντας έναν σταθερό χάρτη. Σε έναν μικροελεγκτή με Cortex-M0+ επεξεργαστή τα πιο συνηθισμένα χαρακτηριστικά που μπορεί κανείς να συναντήσει αναφορικά με τη χαρτογράφηση της μνήμης είναι:

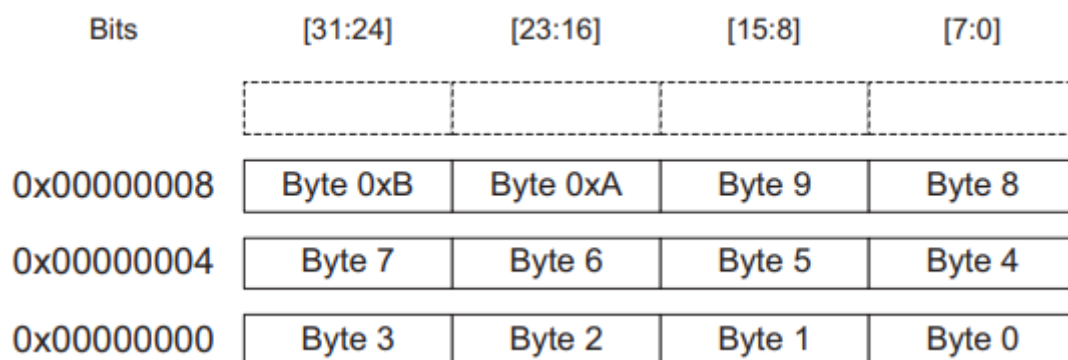
- Flash Memory, για την αποθήκευση κώδικα
- Εσωτερική μνήμη SRAM
- Εξωτερικά περιφερειακά
- Εξωτερική μνήμη

Παρ' όλα αυτά η χρησιμότητα της μνήμης είναι πολύ ευέλικτη, καθώς ένας σχεδιαστής μπορεί να συνδυάσει πράγματα και να διαφοροποιήσει την παραπάνω αναφορά στην τυπική χαρτογράφηση μνήμης. Αξίζει να σημειωθεί ότι κάθε περιοχή μνήμης έχει μια συγκεκριμένη χρησιμότητα και η συμπεριφορά της εξαρτάται ανάλογα με το ποια περιοχή της προσπαθούμε να προσπελάσουμε. Ο διευθυνσιοδοτούμενος χώρος της μνήμης χωρίζεται σε οκτώ κομμάτια, όπου το κάθε ένα έχει μέγεθος 512Mbytes. Βλέποντας την εικόνα 5, η περιοχή διεύθυνσης Code, εμπεριέχει τον κώδικα του προγράμματος, αλλά και τον Vector Table στην αρχική διεύθυνση 0x00000000. Συνεχίζοντας, έχουμε την περιοχή μνήμης SRAM, όπου εκεί αποθηκεύονται όλα τα δεδομένα μας, αλλά και άλλες περιοχές όπως είναι εκείνης των περιφερειακών, της μνήμης RAM, φτάνοντας μέχρι το ανώτερο κομμάτι στον χάρτη που είναι το Reserved Memory Space. Είναι σημαντικό να αναφερθεί ότι κάποιες περιοχές μνήμης μοιράζονται κοινές ιδιότητες. Για παράδειγμα, στην SRAM μπορούμε να αποθηκεύσουμε κώδικα προγράμματος όπως και στην περιοχή Code, όμως στην αντίστοιχη των περιφερειακών δεν γίνεται. Ωστόσο, στην περιοχή των περιφερειακών μπορούμε να αποθηκεύσουμε δεδομένα όπως στην SRAM.



Εικόνα 5 Ο χάρτης της μνήμης

Για την κωδικοποίηση των bytes χρησιμοποιείται η μορφή Little Endian όπου σε μια λέξη το byte με την μικρότερη διεύθυνση είναι το λιγότερο σημαντικό της. Ένα παράδειγμα της συγκεκριμένης μορφής φαίνεται στην παρακάτω εικόνα 6 για μια 32-bit μνήμη.



Εικόνα 6 Η μορφή Little Endian

Όπως έχει αναφερθεί, ο Cortex-M0+ επεξεργαστής χρησιμοποιεί την αρχιτεκτονική ARMv6-M, από την οποία κληρονομεί κάποιες ιδιότητες αλλά και τύπους μνήμης για τις διάφορες περιοχές στον χάρτη της. Ο κύριος λόγος είναι ότι ο συγκεκριμένος επεξεργαστής χρησιμοποιείται από διάφορες συσκευές με μεγάλα συστήματα μνήμης και στόχος είναι η σωστή επικοινωνία μεταξύ τους. Ωστόσο, οι ιδιότητες της μνήμης μπορούν να επηρεάσουν τα δεδομένα αλλά και διάφορες εντολές στην πρόσβασή τους τόσο στην μνήμη όσο και στα περιφερειακά. Κάθε ιδιότητα έχει και μια συγκεκριμένη λειτουργία:

- **Executable:** Ενεργοποιείται κάθε φορά που προσπαθούμε να προσπελάσουμε μια συγκεκριμένη περιοχή μνήμης, με σκοπό να ελεγχθεί αν η εκτέλεση του προγράμματος επιτρέπεται εκεί. Σε περίπτωση που δεν επιτρέπεται χαρακτηρίζεται ως eXecute Never (XN).
- **Buffer able:** Αφορά την ενέργεια εγγραφής προς την μνήμη. Αποθηκεύει τα δεδομένα σε ένα buffer και επιτρέπει στον μικροελεγκτή να προχωρήσει στην εκτέλεση νέας εντολής.
- **Cache able:** Αφορά την ενέργεια ανάγνωσης, όταν για παράδειγμα λαμβάνονται δεδομένα από την μνήμη, αποθηκεύονται προσωρινά στην κρυφή μνήμη με σκοπό όταν ο μικροελεγκτής χρειαστεί ξανά τα συγκεκριμένα δεδομένα να τα ανακτήσει από την κρυφή μνήμη για να επιταχύνει την διαδικασία της εκτέλεσης.
- **Shareable:** Η συγκεκριμένη ιδιότητα αφορά την πολλαπλή και ταυτόχρονη προσπέλαση μιας συγκεκριμένης περιοχής μνήμης από περισσότερους από έναν επεξεργαστές.

Έχοντας ως βάση τις παραπάνω ιδιότητες αναπτύχθηκαν και κάποιοι τύποι μνήμης σε αυτή την αρχιτεκτονική. Οι δύο κατηγορίες ενημερώνονται από τον επεξεργαστή και επικοινωνούν με το υπόλοιπο σύστημα μέσω του διαύλου AHB. Οι τύποι δεδομένων για την μνήμη του μικροελεγκτή είναι τρεις με τον πρώτο να χαρακτηρίζεται ως Normal memory και μπορεί να έχει μόνο μια από δύο ιδιότητες από όσες προαναφέρθηκαν, συγκεκριμένα την Cacheable και την Shareable ιδιότητα. Ο συγκεκριμένος τύπος μνήμης μπορεί να παρέχει επιπλέον ιδιότητες όπως Write Through (WT), Write Back και Write Allocate (WBWA) μόνο στην περίπτωση που έχει την cacheable ιδιότητα. Συνεχίζοντας ο δεύτερος τύπος μνήμης είναι ο Device memory που δεν μπορεί να έχει την ιδιότητα cacheable, αλλά μόνο την ιδιότητα shareable. Τέλος η τρίτη κατηγορία ονομάζεται Strongly Ordered memory η οποία δεν έχει την ιδιότητα της κρυφής μνήμης αλλά και την ταυτόχρονη προσπέλαση που επιτρέπει η shareable ιδιότητα πράγμα που επηρεάζει την μεταφορά δεδομένων από και προς την συγκεκριμένη περιοχή μνήμης. Ένα παράδειγμα μια τυπικής διαχείρισης όσον αναφέρθηκαν παραπάνω στις περιοχές της μνήμης είναι το παρακάτω.

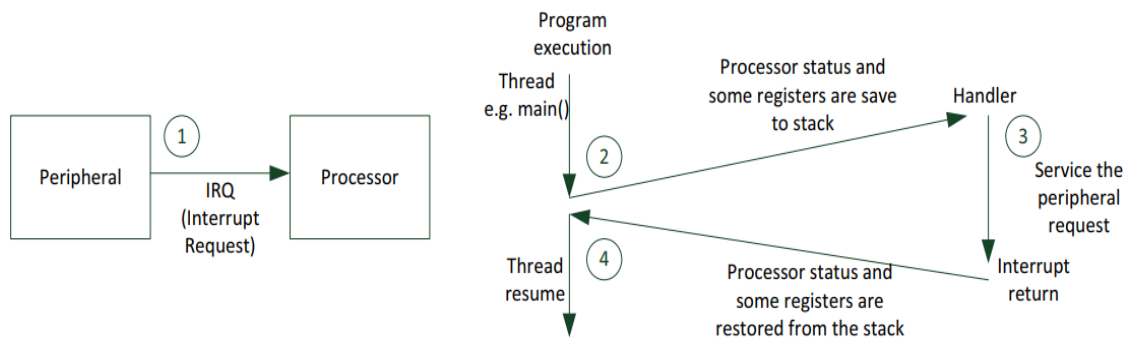
Address	Region	Memory type	Cache	XN	Shareable	Descriptions
0x00000000— 0x1FFFFFFF	CODE	Normal	WT	—	—	Memory for program code including vector table
0x20000000— 0x3FFFFFFF	SRAM	Normal	WBWA	—	—	SRAM, typically used for data and stack memory
0x40000000— 0x5FFFFFFF	Peripheral	Device	—	XN	—	Typically used for on-chip devices
0x60000000— 0x7FFFFFFF	RAM	Normal	WBWA	—	—	Normal memory with Write Back, Write Allocate cache attributes
0x80000000— 0x9FFFFFFF	RAM	Normal	WT	—	—	Normal memory with Write Through cache attributes
0xA0000000— 0xBFFFFFFF	Device	Device	—	XN	S	Shareable device memory
0xC0000000— 0xDFFFFFFF	Device	Device	—	XN	—	Nonshareable device memory
0xE0000000— 0xE00FFFFF	PPB	Strongly ordered	—	XN	S	Internal Private Peripheral Bus
0xE0100000— 0xFFFFFFF	Reserved	Reserved	—	—	—	Reserved (Vendor-specific usage)

Εικόνα 7 Ιδιότητες και τύποι μνήμης

2.2.3 Εξαιρέσεις και διακοπές συστήματος

Οι δύο όροι που αναφέρονται ως διακοπή (Interrupt) και εξαίρεση (Exception) είναι στην ουσία σήματα που αποστέλλονται προς τον επεξεργαστή, με σκοπό να προκαλέσουν αλλαγή στην ροή του κώδικα που εκτελείται. Πιο αναλυτικά, τα σήματα διακοπών διακρίνονται σε δύο κατηγορίες, εκείνα που προέρχονται από τον επεξεργαστή και ονομάζονται exceptions και εκείνα που προέρχονται από τα ίδια τα περιφερειακά του ολοκληρωμένου συστήματος που ονομάζονται interrupts. Η λειτουργία ενός σήματος διακοπής χαρακτηρίζεται από κάποια συγκεκριμένα βήματα όπως μπορούμε να διακρίνουμε και στην εικόνα 8 η οποία είναι από το βιβλίο του

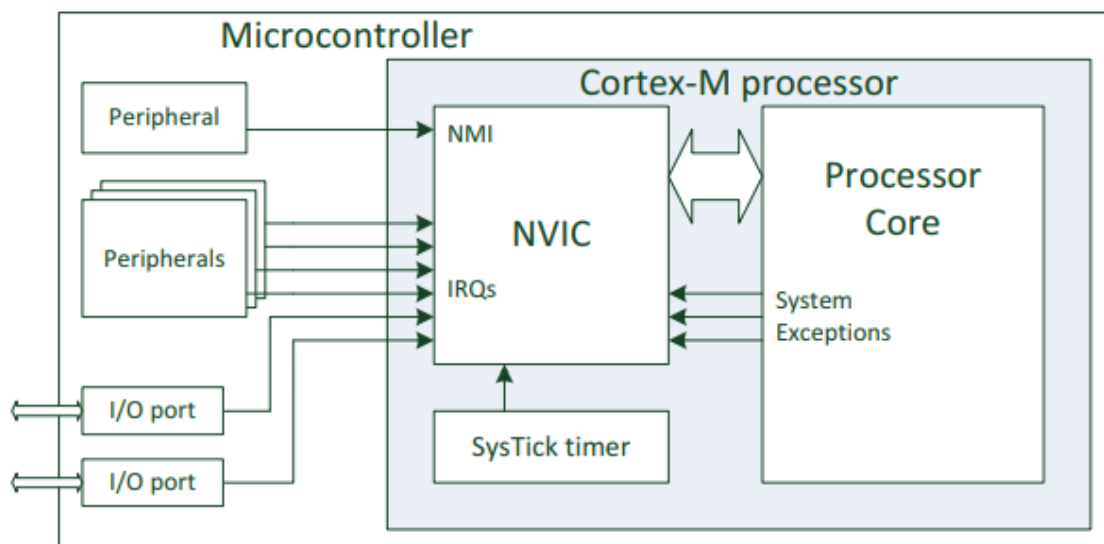
Joseph Yiu (The Definitive Guide to ARM® Cortex®-M0 and Cortex-M0+ Processors). Εμβαθύνοντας, ένα περιφερειακό ξεκινά μια αίτηση διακοπής προς τον επεξεργαστή. Η ενέργεια αυτή χαρακτηρίζεται IRQ (Interrupt Request) και σχετίζεται άμεσα με τα περιφερειακά του ίδιου του ολοκληρωμένου συστήματος καθώς επίσης και με τα εξωτερικά σήματα διακοπής που μπορούν να προσβάλλουν το σύστημα μέσω των περιφερειακών γενικού σκοπού GPIO (General Purpose Input Output ports). Από την στιγμή που ο επεξεργαστής λάβει το σήμα, αποθηκεύει στην στοίβα την διαδικασία κώδικα που εκτελείται εκείνη την στιγμή με όλες τις σημαντικές πληροφορίες όπως τον PC (Program Counter), τον xPSR (Program Status Register) και άλλα. Στην συνέχεια, αναζητά την διεύθυνση του κώδικα προγράμματος Interrupt Handler και εκτελεί την συγκεκριμένη διαδικασία. Τέλος, επαναφέρει ότι έχει αποθηκευτεί στην στοίβα και συνεχίζει την ροή εκτέλεσης από το σημείο που είχε διακοπεί.



Εικόνα 8 Η λειτουργία σήματος διακοπής.

2.2.4 Ελεγκτής διακοπών NVIC

Ο πυρήνας Cortex-M0+ χρησιμοποιεί έναν ενσωματωμένο ελεγκτή διακοπών που επιτρέπει μέσω λογισμικού την διαχείριση των σημάτων διακοπής (exceptions-interrupts). Υποστηρίζει μέχρι και τριάντα δύο εισόδους για σήματα διακοπής από εξωτερικά περιφερειακά, μια μη αποκρύψιμη είσοδο NMI (Non Maskable Interrupt) αλλά και σήματα εξαιρέσεων που προέρχονται από τον επεξεργαστή. Γενικότερα, ο ελεγκτής διακοπών είναι υπεύθυνος για την διαχείριση όλων των σημάτων διακοπής από οποιοδήποτε μέσω προέρχονται, όπως απεικονίζει η εικόνα 9.



Εικόνα 9 Αρχιτεκτονική του ελεγκτή διακοπών, στον πυρήνα Cortex-M0+

Για τον σκοπό λοιπόν της διαχείρισης χρησιμοποιούνται κάποιοι καταχωρητές που οργανώνονται στην μνήμη χρησιμοποιώντας τον χώρο ελέγχου συστήματος SCS για την αποθήκευση των διευθύνσεών τους. Τα κύρια χαρακτηριστικά του ελεγκτή διακοπών βασίζονται σε τέσσερις λειτουργίες. Αρχικά, προσφέρει ευέλικτη διαχείριση στα σήματα διακοπής, καθώς επιτρέπει την ενεργοποίηση ή την επενεργοποίηση τόσο ενός τέτοιου σήματος όσο και της κατάστασής τους όταν βρίσκονται σε εκκρεμότητα. Έπειτα, διαθέτει ενσωματωμένη υποστήριξη που ελέγχει την θέση προτεραιότητας σε ορισμένα σήματα διακοπής καθώς κάποια από αυτά έχουν σταθερή θέση όσον αφορά την προτεραιότητά τους και δεν μπορούν να αλλαχθούν. Καθιστά δυνατή την πρόσβαση μέσω λογισμικού στην κατάσταση εκκρεμότητας των σημάτων διακοπής προσφέροντας στον προγραμματιστή μεγαλύτερη άνεση στην διαχείρισή τους. Τέλος, διαθέτει έναν ξεχωριστό καταχωρητή που ονομάζεται PRIMASK, ο οποίος ελέγχει τα σήματα διακοπής απενεργοποιώντας τα όλα εκτός από το NMI και το HardFault. Τέλος, αξίζει να σημειωθεί πως όλα τα σήματα αυτού του είδους έχουν διανυσματική μορφή, γεγονός που αιτιολογεί πως ο επεξεργαστής λειτουργεί όταν αποφασίσει να επιλέξει την αποδοχή ενός σήματος IRQ. Ο λόγος γίνεται για το λεγόμενο Vector Table που προσφέρει ως πληροφορία την αρχική διεύθυνση ενός τέτοιου σήματος προκειμένου ο επεξεργαστής να εκτελέσει το αντίστοιχο exception ή interrupt Handler.

2.2.4.1 Ορισμός προτεραιότητας στα σήματα διακοπών

Προκειμένου να γνωρίζουμε πότε ένα σήμα που προκαλεί διακοπή στην ροή του προγράμματος (Εξαίρεσης/Διακοπής), θα περιμένει να εκτελεστεί (κατάσταση εκκρεμότητας) ή θα τεθεί σε εκτέλεση, ο πυρήνας Cortex-M0+ έχει ορίσει μια

αριθμημένη σειρά προτεραιότητας. Όσο πιο μικρός είναι ο αριθμός της προτεραιότητας που έχει ένα τέτοιο σήμα τόσο πιο μεγάλη είναι η προτεραιότητά του για εκτέλεση. Στον πυρήνα υποστηρίζονται τρία μη προγραμματίσιμα επίπεδα που κατέχουν τις υψηλότερες θέσεις στα επίπεδα προτεραιότητας και αφορούν τις εξαιρέσεις του συστήματος Reset, NMI, HardFault, καθώς επίσης και άλλα τέσσερα προγραμματίσιμα επίπεδα για κάθε σήμα διακοπής είτε εξαίρεσης.

2.2.5 Οι ταλαντωτές του συστήματος

Όσον αφορά την διαχείριση του χρονισμού στο λειτουργικό σύστημα του επεξεργαστή, διατίθενται εσωτερικοί RC ταλαντωτές που μπορούν να ενεργοποιήσουν τον εσωτερικό βρόγχο κλειδώματος PLL (Phase Lock Loop). Ωστόσο, αν αναζητούμε ακρίβεια και σταθερότητα στον χρονισμό του επεξεργαστή τότε η πιο αξιόπιστη λύση είναι οι εξωτερικοί κρύσταλλοι σε σχέση με τους εσωτερικούς ταλαντωτές. Έτσι λοιπόν υπάρχουν τέσσερις διαθέσιμες πηγές ρολογιού για το σύστημα αυτό. Ξεκινώντας με έναν υψηλής ταχύτητας εξωτερικό ταλαντωτή τον HSE (High Speed External) και συνεχίζοντας με τον υψηλής ταχύτητας εσωτερικό ταλαντωτή HSI16(High Speed Internal) με λειτουργία 16 MHz, το ρολόι PLL (Phase Lock Loop) και τέλος το εσωτερικό ρολόι MSI(Multi speed internal), που είναι χρήσιμο κυρίως ως ρολόι του συστήματος όταν το σύστημά μας επανέρχεται από μια κατάσταση χαμηλής ενέργειας ή από κάποια επανεκκίνηση. Εμβαθύνοντας, υπάρχουν ακόμη δύο εσωτερικοί RC ταλαντωτές όπως ο εσωτερικός χαμηλής ταχύτητας LSI (Low Speed Internal) με συχνότητα 37kHz, που χρησιμοποιείται κυρίως για τον επιτηρητή του συστήματος watchdog, καθώς επίσης και ο εξωτερικός ταλαντωτής χαμηλής ταχύτητας LSE(Low Speed External) με συχνότητα 32.768 kHz, που καθορίζει το ρολόι πραγματικού χρόνου RTC.

2.2.5.1 HSE (Εξωτερικός ταλαντωτής υψηλής ταχύτητας)

Η συγκεκριμένη πηγή ρολογιού μπορεί να ενεργοποιηθεί από δύο διαφορετικές καταστάσεις. Η πρώτη είναι με την μορφή ενός κρυσταλλικού/κεραμικού συντονιστή και η δεύτερη περίπτωση βασίζεται σε παλμούς ρολογιού προερχόμενες από τον χρήστη. Ξεκινώντας με την πρώτη περίπτωση, αξίζει να αναφερθεί ότι, ο εξωτερικός κεραμικός κρύσταλλος πρέπει να λειτουργεί ανάμεσα στις συχνότητες 1 έως 24 MHz, ώστε να λειτουργεί κατάλληλα με το ρολόι PLL, καθώς εκείνο δέχεται ακέραια πολλαπλάσια της συχνότητας του HSE. Όσον αφορά την δεύτερη κατηγορία, η μέγιστη συχνότητα που μπορεί να έχει ο παλμός ρολογιού είναι τα 32MHz με τον κύκλο εργασίας του να βρίσκεται στα 50%. Η εμφάνιση του παλμού σε ειδικά μηχανήματα μπορεί να είναι ημιτονοειδής, τετραγωνική ή τριγωνική.

2.2.5.2 HSI (Εσωτερικός ταλαντωτής υψηλής ταχύτητας)

Η περίπτωση του εσωτερικού RC ταλαντωτή υψηλής ταχύτητας έχει αρκετά πλεονεκτήματα σε σχέση με τον εξωτερικό ταλαντωτή HSE. Τόσο το μηδαμινό κόστος κατασκευής του (δεν υπάρχουν εξωτερικά κομμάτια), όσο και το γεγονός ότι μπορεί να χρησιμοποιηθεί ως ρολόι συστήματος απευθείας ή να εισέλθει στο PLL, το κάνουν να ξεχωρίζει. Ωστόσο, το βασικό του μειονέκτημα εντοπίζεται στην μη αποδοτική ακρίβεια συγκριτικά με τον εξωτερικό ταλαντωτή HSE. Η μέγιστη συχνότητά του είναι τα 16 MHz και η διαχείρισή του καθίσταται αρκετά εύκολη εξαιτίας του καταχωρητή που διαθέτει RCC_CR θέτοντας και μηδενίζοντας το bit του με όνομα HSI16ON.

2.2.5.3 MSI (Εσωτερικός ταλαντωτής πολλαπλών ταχυτήτων)

Ο εσωτερικός RC ταλαντωτής πολλαπλών ταχυτήτων έχει επτά διαθέσιμες συχνότητες λειτουργίας: 65.536 kHz, 131.072 kHz, 262.144 kHz, 524.288 kHz, 1.048 MHz, 2.097 MHz (default value) και 4.194 MHz. Κύριο χαρακτηριστικό του είναι ότι παρέχει στο σύστημα παλμούς ρολογιού με χαμηλή κατανάλωση αλλά και το γεγονός ότι το κόστος του είναι σχεδόν μηδενικό, αφού δεν υπάρχουν εξωτερικά κομμάτια. Μπορεί να χρησιμοποιηθεί ως ρολόι συστήματος απευθείας όπως και το HSI16 και είναι κατάλληλο για την αφύπνιση του μικροελεγκτή από καταστάσεις χαμηλής κατανάλωσης και επανεκκίνησης. Αξίζει να αναφερθεί ότι στην περίπτωση που ο εξωτερικός RC ταλαντωτής υψηλής ταχύτητας HSE παρουσιάσει κάποιο πρόβλημα στην λειτουργία του μπορεί να χρησιμοποιηθεί ως βοηθητικό ρολόι για το σύστημα στην αντίστοιχη θέση του.

2.2.5.4 PLL (Εσωτερικός βρόγχος κλειδώματος φάσης)

Ο εσωτερικός βρόγχος κλειδώματος φάσης μπορεί να τροφοδοτηθεί από δύο πηγές ρολογιού μόνο. Αυτές είναι ο εξωτερικός RC ταλαντωτής υψηλής ταχύτητας HSE και ο εσωτερικός ταλαντωτής υψηλής ταχύτητας των 16MHz HSI16. Μπορεί να κατευθύνει το ρολόι του συστήματος, έχοντας ως σημείο αναφοράς το συγκεκριμένο εύρος συχνοτήτων με το οποίο μπορεί να λειτουργήσει και αυτό είναι από 2 έως 24 MHz. Αν χρησιμοποιηθεί ο εξωτερικός ταλαντωτής HSE, η ελάχιστη συχνότητα που μπορεί να έχει το ρολόι του συστήματος είναι τα 2 MHz. Για οποιαδήποτε ενέργεια αναφορικά με τη διαμόρφωση του ρολογιού πρέπει πρώτα να είναι απενεργοποιημένο και έπειτα να πραγματοποιηθούν οι απαραίτητες τροποποιήσεις.

2.2.6 Λειτουργίες χαμηλής κατανάλωσης

Ο επεξεργαστής εξ' ορισμού λειτουργεί σε μια κανονική κατάσταση που ονομάζεται run mode, έπειτα από κάποια επανεκκίνηση. Στα σύγχρονα συστήματα που δημιουργούνται απαιτείται να υπάρχει χαμηλή κατανάλωση στην ισχύ τους και αυτό μπορούμε γρήγορα και εύκολα να το πετύχουμε στον επεξεργαστή μας με δύο απλούς τρόπους. Ο πρώτος τρόπος βασίζεται στην επιβράδυνση του ρολογιού στο σύστημά μας και ο δεύτερος επιτυγχάνεται απενεργοποιώντας όσα περιφερειακά δεν μας είναι χρήσιμα. Παρ' όλα αυτά ο επεξεργαστής διαθέτει πέντε καταστάσεις χαμηλής κατανάλωσης που προσφέρουν αρκετά χαρακτηριστικά και στις περισσότερες από αυτές δεν απαιτείται ο επεξεργαστής να συνεχίζει να βρίσκεται σε κατάσταση κανονικής λειτουργίας όταν είναι ενεργοποιημένη μια από αυτές. Έτσι, λοιπόν, ο προγραμματιστής καλείται να επιλέξει την κατάλληλη λειτουργία χαμηλής κατανάλωσης για το σύστημά του σύμφωνα με τις εκάστοτε προδιαγραφές που προσφέρει η κάθε μια, όπως ο χρόνος που χρειάζεται το σύστημα για να επανέλθει στην κανονική (run mode) κατάσταση, η κατανάλωση ενέργειας αλλά και οι διάφοροι τρόποι που απαιτούνται για την επίτευξη του σκοπού.

Mode name	Entry	Wakeup	Effect on V _{CORE} domain clocks	Effect on V _{DD} domain clocks	Voltage regulator
Low-power run	LPSSDR and LPRUN bits + Clock setting	The regulator is forced in Main regulator (1.8 V)	None	None	In low-power mode
Sleep (Sleep now or Sleep-on-exit)	WFI or Return from ISR	Any interrupt	CPU CLK OFF no effect on other clocks or analog clock sources	None	ON
	WFE	Wakeup event			
Low-power sleep (Sleep now or Sleep-on-exit)	LPSSDR bits + WFI or Return from ISR	Any interrupt	CPU CLK OFF no effect on other clocks or analog clock sources, Flash CLK OFF	None	In low-power mode
	LPSSDR bits + WFE	Wakeup event			
Stop	PDDS, LPSSDR bits + SLEEPDEEP bit + WFI, Return from ISR or WFE	Any EXTI line (configured in the EXTI registers, internal and external lines)	All V _{CORE} domain clocks OFF	HSI16 ⁽¹⁾ , HSE and MSI oscillators OFF	In low-power mode
Standby	PDDS bit + SLEEPDEEP bit + WFI, Return from ISR or WFE	WKUP pin rising edge, RTC alarm (Alarm A or Alarm B), RTC Wakeup event, RTC tamper event, RTC timestamp event, external reset in NRST pin, IWDG reset			OFF

Εικόνα 10 Λειτουργίες χαμηλής κατανάλωσης

2.2.6.1 Low Power Run Mode

Στην συγκεκριμένη κατάσταση χαμηλής κατανάλωσης όλα τα περιφερειακά του συστήματος συνεχίζουν να έχουν τις ίδιες ρυθμίσεις που είχανε και στην κανονική κατάσταση run mode, με τη διαφορά ότι ο αριθμός τους είναι περιορισμένος. Για να εισέλθει το σύστημα σε αυτή την κατάσταση αρκεί η ενεργοποίηση του εσωτερικού ταλαντωτή RC, πολλαπλών ταχυτήτων MSI με μέγιστη συχνότητα λειτουργίας 131 kHz και εκτέλεση αυτής της ενέργειας από την μνήμη SRAM ή Flash.

2.2.6.2 Sleep Mode

Έχοντας ενεργοποιημένη την συγκεκριμένη κατάσταση μπορούμε να πετύχουμε 1 mA μείωση στην κατανάλωση του συστήματός μας. Ο επεξεργαστής είναι ο μόνος που αδρανοποιείται και όλα τα περιφερειακά συνεχίζουν να λειτουργούν κανονικά. Για την ενεργοποίηση της κατάσταση Sleep Mode, διατίθενται δύο εντολές με τις ακόλουθες ονομασίες: WFI (Wait For Interrupt), WFE (Wait For Event). Θέτοντας αυτές τις εντολές ο επεξεργαστής αδρανοποιείται και περιμένει την αφύπνισή του. Στο σημείο αυτό βρίσκεται και η κύρια διαφορά των δύο εντολών. Με την πρώτη, ο επεξεργαστής περιμένει μια αίτηση διακοπής για την αφύπνιση του (IRQ), ενώ στην δεύτερη περιμένει ένα γεγονός (event). Αξίζει να σημειωθεί ότι, όταν ο επεξεργαστής βρίσκεται σε αδρανοποίηση, έχοντας ενεργοποιημένη την κατάσταση Sleep Mode, απενεργοποιεί το εσωτερικό ρολόι του συστήματος, οπότε ο κώδικας του προγράμματος σταματά. Ωστόσο, όλες οι περιφερειακές συσκευές είναι ενεργές ώστε να μπορέσουν με έναν από τους δύο παραπάνω τρόπος να τον ενεργοποιήσουν και πάλι.

2.2.6.3 Low Power Sleep Mode

Η κατάσταση Low Power Sleep Mode είναι ένας συνδυασμός από την κατάσταση Low Power Run Mode και Sleep Mode κατά την οποία τόσο ο επεξεργαστής τίθεται σε αδράνεια όσο και ο εσωτερικός ρυθμιστής τάσης τίθεται σε κατάσταση χαμηλής κατανάλωσης. Για το γεγονός λοιπόν ότι η συγκεκριμένη κατάσταση είναι ένας συνδυασμός καταστάσεων, αυτόματα κληρονομεί όλα τα χαρακτηριστικά των Low Power Run Mode και Sleep Mode.

2.2.6.4 Stop Mode

Μια από τις πιο αποδοτικές καταστάσεις όσον αφορά την μείωση της κατανάλωσης είναι η κατάσταση Stop Mode. Η ενεργοποίηση της κατάστασης μπορεί να επιτευχθεί από δύο λειτουργίες. Η πρώτη αφορά το ρολόι πραγματικού χρόνου RTC και η δεύτερη είναι αυτόνομη χωρίς την χρησιμοποίηση του ρολογιού RTC. Τόσο στην μια

περίπτωση όσο και στην άλλη ο επεξεργαστής πετυχαίνει μια από τις μεγαλύτερες αποδόσεις χαμηλής κατανάλωσης, διατηρώντας ωστόσο την μνήμη RAM και τους καταχωρητές ενεργούς και στις δύο περιπτώσεις, έχοντας βέβαια στην πρώτη ενεργοποιημένο και το ρολόι πραγματικού χρόνου. Ξεκινώντας με την κατάσταση Stop Mode και με το ρολόι πραγματικού χρόνου ενεργοποιημένο, όλοι οι RC ταλαντωτές του συστήματος είναι απενεργοποιημένοι, διατηρώντας ωστόσο ενεργούς μόνο το εξωτερικό ρολόι χαμηλής ταχύτητας LSE (Low Speed External) και το επίσης χαμηλής ταχύτητας εσωτερικό ρολόι LSI (Low Speed Internal). Ταυτόχρονα ο ρυθμιστής τάσης είναι σε κατάσταση χαμηλής κατανάλωσης. Η συσκευή μπορεί να αφύπνιστεί από οποιαδήποτε μονάδα εξωτερικών διακοπών(EXTI), στέλνοντας μια αίτηση διακοπής στον επεξεργαστή IRQ. Η συγκεκριμένη πηγή μπορεί να είναι κάποια είσοδος στα περιφερειακά γενικού σκοπού GPIO's. Αν για παράδειγμα είναι μια εξωτερική συσκευή που έχει τεθεί υπεύθυνη για την αφύπνιση μπορεί να είναι κάποια έξοδος PVD ή κάποιο event. Στην αντίθετη περίπτωση έχουμε το ρολόι πραγματικού χρόνου RTC, tamper, διάφορα γεγονότα που μπορούν να προκαλέσουν αφύπνιση καθώς επίσης και περιφερειακά επικοινωνίας όπως USB, USART, I2C, LPUART και LPTIMER. Στην δεύτερη περίπτωση Sleep Mode που δεν υπάρχει το RTC ισχύουν όσα προαναφέρθηκαν, χωρίς να λαμβάνεται υπόψιν το ρολόι πραγματικού χρόνου ως μέσω αφύπνισης του επεξεργαστή.

2.2.6.5 Standby Mode

Η κατάσταση Standby Mode είναι η πιο αποδοτική μέθοδος στην μείωση κατανάλωσης. Μπορούμε και σε αυτή την κατάσταση να χρησιμοποιήσουμε το ρολόι πραγματικού χρόνου ως μέσω αφύπνισης του επεξεργαστή, αλλά γίνεται και χωρίς αυτό. Τόσο στην μια περίπτωση όσο και στην άλλη, ο εσωτερικός ρυθμιστής τάσης είναι απενεργοποιημένος όπως επίσης και οι περισσότερες πηγές ρολογιού. Στο σημείο αυτό, διακρίνεται και η κύρια διαφορά των δύο περιπτώσεων καθώς αν χρησιμοποιηθεί το ρολόι πραγματικού χρόνου, τότε διατηρούνται σταθερές οι πηγές ρολογιού LSI και LSE, σε αντίθεση με την αυτόνομη περίπτωση που είναι όλες οι πηγές απενεργοποιημένες. Αφού ο επεξεργαστής εισέλθει σε αυτή την λειτουργία εξοικονόμησης ενέργειας τα δεδομένα των καταχωρητών και της μνήμης RAM χάνονται εκτός από τους καταχωρητές που βρίσκονται στα: wakeup logic, IWDG, RTC, LSI, LSE και στους καταχωρητές RCC_CSR. Η αφύπνιση του επεξεργαστή από την συγκεκριμένη κατάσταση γίνεται μετά από 60 ms χρόνου, αφότου ενεργοποιηθεί το εξωτερικό pin επανεκκίνησης NRST ή αν εμφανιστεί μια ανοδική ακμή του ρολογιού από τα τρία pin WKUP. Ωστόσο στην περίπτωση που το RTC είναι ενεργοποιημένο μπορούν να θεωρηθούν κατάλληλα

μέσα για αφύπνιση του συστήματος όσα προαναφέρθηκαν ή και τα ακόλουθα: RTC alarm, RTC tamper event, RTC timestamp event, RTC wake up event και από μια επανεκκίνηση του επιτηρητή IWDG.

2.3 Περιφερειακά Γενικού Σκοπού

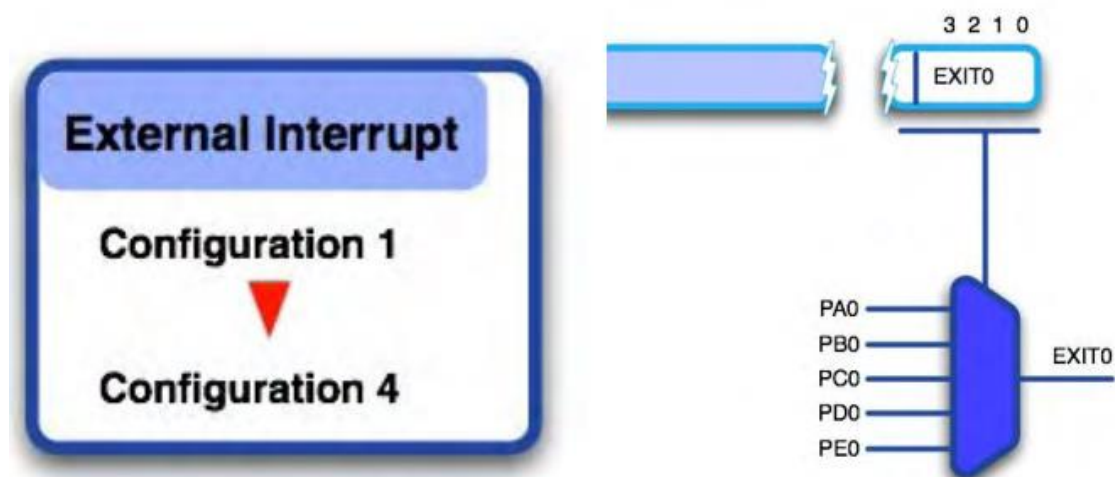
Τα περιφερειακά των μικροελεγκτών είναι κομμάτια της συσκευής που έχουν ως κύριο στόχο τους την βοήθεια του μικροελεγκτή για την επίτευξη μιας συγκεκριμένης εργασίας. Μερικά κοινά περιφερειακά είναι τα εξής: μονάδα ενσωματωμένων εξωτερικών διακοπών, είσοδοι/έξοδοι γενικού σκοπού, διάφορα περιφερειακά επικοινωνίας, οι χρονιστές γενικού σκοπού και άλλα.

2.3.1 Είσοδοι/Έξοδοι γενικού σκοπού GPIO's

Ο STM32 διαθέτει πέντε θύρες εισόδων/εξόδων που είναι περιφερειακά κυκλώματα και παρέχουν ένα πλήθος δεκαέξι ακροδεκτών (pins) γενικού σκοπού. Για κάθε θύρα υπάρχει ένας καταχωρητής κατεύθυνσης και ένας καταχωρητής δεδομένων, με τον πρώτο να καθορίζει τους ακροδέκτες της θύρας που θα λειτουργούν ως είσοδοι ή ως έξοδοι. Οι ακροδέκτες που λειτουργούν ως έξοδοι οδηγούνται στον καταχωρητή δεδομένων μέσω μιας τιμής, ενώ αντίστοιχα οι ακροδέκτες που δρουν ως είσοδοι απεικονίζονται στα bit του καταχωρητή δεδομένων, όταν εκείνος διαβαστεί από την κεντρική μονάδα του επεξεργαστή ΚΜΕ. Όλες οι θύρες διαθέτουν από δύο καταχωρητές μεγέθους τριάντα δύο bit οι οποίοι είναι υπεύθυνοι για τον καθορισμό των λειτουργιών κάθε θύρας. Αυτόματα αυτό σημαίνει ότι σε κάθε θύρα αναλογούν τέσσερα bits εκ των οποίων τα πρώτα δύο διαμορφώνουν την στάση της θύρας στο σύστημα, αν θα λειτουργεί δηλαδή ως είσοδος ή ως έξοδος και τα υπόλοιπα είναι υπεύθυνα για τον καθορισμό της λειτουργικής συμπεριφοράς κάθε θύρας, δηλαδή το πως θα οδηγεί τα αντίστοιχα χαρακτηριστικά των ακροδεκτών. Το γεγονός αυτό, προσφέρει οκτώ τρόπους λειτουργίας στους ακροδέκτες όπως για παράδειγμα: open drain output, push-pull output, input floating, input pull-up, input pull-down, AF push-pull/open-drain). Αξίζει να σημειωθεί ότι δύναται η δυνατότητα προγραμματισμού της ταχύτητας εξόδου στις θύρες με τις μέγιστες επιλογές να είναι 2MHz, 10MHz και 50MHz. Τέλος, διατίθενται και οι καταχωρητές με επαναπρογραμματιζόμενη λειτουργία στους ακροδέκτες, έτσι ώστε να επιτευχθεί η εξυπηρέτηση κάποιων περιφερειακών επικοινωνίας όπως είναι τα αντίστοιχα: SPI, USART, I2C, CAN και timers.

2.3.2 μονάδα εξωτερικών διακοπών συστήματος EXT1

Η συγκεκριμένη μονάδα διαθέτει δεκαεννιά γραμμές διακοπών οι οποίες είναι άμεσα συνδεδεμένες με τον ενσωματωμένο ελεγκτή διακοπών. Είναι υπεύθυνη για τον έλεγχο της κατανάλωσης ενέργειας του συστήματος, έχοντας μεγάλο πλεονέκτημα ότι δεν απαιτείται κάποια πηγή ρολογιού για το χρονισμό των γραμμών της μονάδας. Αυτό καθιστά δυνατή την ενέργεια αφύπνισης ενός επεξεργαστή ακόμη και αν το ρολόι του συστήματος είναι απενεργοποιημένο, καθώς όλες οι γραμμές της εξωτερικής μονάδας διακοπών μπορούν να προκαλέσουν διακοπή συστήματος. Οι δεκαέξι από αυτές είναι συνδεδεμένες με τις εισόδους/εξόδους γενικού σκοπού (GPIO's), έχοντας την ικανότητα να προκαλέσουν διακοπή σε ανοδική/καθοδική ακμή ρολογιού ή ακόμη και στα δύο ταυτόχρονα. Η συνέχεια βρίσκει τις επόμενες τρεις γραμμές EXT1 να είναι συνδεδεμένες με την αφύπνιση USB, το ρολόι πραγματικού χρόνου καθώς επίσης και με την μονάδα ανίχνευσης τάσης (Power Voltage Detect). Επίσης για τις υπόλοιπες γραμμές 5 έως 9 και 10 έως 15 διατίθενται δύο διαφορετικά διανύσματα διακοπών. Οι δεκαέξι γραμμές που είναι άμεσα συνδεδεμένες με τις εισόδους/εξόδους γενικού σκοπού μπορούν να χρησιμοποιηθούν με αρκετούς συνδυασμούς. Αυτό γίνεται με την βοήθεια συνολικά τεσσάρων καταχωρητών, όπου κάθε γραμμή EXT1 συσχετίζεται σε ένα πεδίο αποτελούμενο από τέσσερα bits. Το συγκεκριμένο πεδίο προσφέρει την δυνατότητα αντιστοίχισης των γραμμών EXT1 σε οποιαδήποτε από τις πέντε θύρες εισόδου/εξόδου, όπως για παράδειγμα η γραμμή EXT1 0 αντιστοιχίζεται με το pin 0 μιας οποιαδήποτε θύρας A, B, C, D, E. Έτσι καθίσταται εύκολη η σύνδεση ενός εξωτερικού pin με μια γραμμή διακοπής, προκειμένου να μπορεί να στείλει την αντίστοιχη αίτηση προς τον επεξεργαστή.



Εικόνα 11 Εξωτερικές γραμμές διακοπών EXTI

2.3.3 Χρονιστές/Μετρητές Γενικού Σκοπού

Οι χρονιστές/μετρητές είναι λογικά κυκλώματα που υποστηρίζουν παράλληλη φόρτιση και η τιμή των οποίων αυξομειώνεται σύμφωνα με τους παλμούς κάποιου ρολογιού. Ο μικροελεγκτής STM32L053R8 διαθέτει τρεις χρονιστές γενικού σκοπού, δύο χρονιστές επιτήρησης με την ονομασία watchdog, επιπλέον ένα χρονιστή/μετρητή χαμηλής κατανάλωσης LPTIM και έναν βασικό χρονιστή/μετρητή, μαζί με εκείνον του συστήματος SysTick timer. Οι χρονιστές/μετρητές είναι 16bit και είναι άμεσα συνδεδεμένοι τις περισσότερες φορές με έναν διαιρέτη συχνότητας, εξίσου 16bit που ονομάζεται prescaler. Η κύρια λειτουργία του είναι ότι προσφέρει την δυνατότητα της μεταβολής της τιμής των μετρητών/χρονιστών κατά ένα, ανά έναν αριθμό παλμών εσωτερικού ή εξωτερικού ρολογιού. Ένα χαρακτηριστικό παράδειγμα είναι αν θέσουμε των Prescaler στην τιμή 1:32, πράγμα που σημαίνει αυτόματα ότι η τιμή του μετρητή/χρονιστή μεταβάλλεται κατά ένα, ανά τριάντα δύο παλμούς ρολογιού. Έτσι, υπάρχει η δυνατότητα μέτρησης μεγάλων χρονικά διαστημάτων, έχοντας ωστόσο μειωμένη ακρίβεια. Ο διαχωρισμός των δύο λογικών κυκλωμάτων είναι αρκετά απλός. Χρονιστής θεωρείται το λογικό κύκλωμα όπου ο μετρητής του μεταβάλλεται σύμφωνα με τους παλμούς ενός εσωτερικού ρολογιού σταθερής συχνότητας. Αντίστοιχα, αν ο μετρητής του λογικού κυκλώματος μεταβάλλεται σύμφωνα με τους παλμούς ενός εξωτερικού ρολογιού τότε λέμε ότι το λογικό αυτό κύκλωμα λειτουργεί ως μετρητής. Η πρόκληση διακοπών του συστήματος είναι ένα φαινόμενο που μπορεί να προκληθεί από αυτού του είδους λογικών κυκλωμάτων λόγω υπερχειλίσης. Αυτό συμβαίνει όταν η τιμή ενός χρονιστή/μετρητή κατά την μείωσή της ή την αύξησή της φτάσει στο μηδέν ή ξεπεράσει την μέγιστη τιμή της. Μια άλλη περίπτωση πρόκλησης διακοπής στο σύστημα

είναι αν η αντίστοιχη τιμή γίνει ίση με μια προκαθορισμένη τιμή, όπου αυτή η περίπτωση διακοπής χρησιμοποιείται για την δημιουργία παλμών με ρυθμιζόμενη συχνότητα και με διάρκεια υψηλής/χαμηλής στάθμης (duty cycle). Αυτό είναι γνωστό και ως φαινόμενο Pulse Width Modulation. Οι χρονιστές/μετρητές διαθέτουν δύο αρκετά σημαντικές λειτουργίες που είναι η Input capture και η Output Compare. Η πρώτη προσφέρει την δυνατότητα της μεταφοράς της τρέχουσας τιμής ενός χρονιστή/μετρητή, όταν αλλάξει η στάθμη ενός ακροδέκτη εισόδου, σε κάποιον άλλο καταχωρητή με κύριο σκοπό την αποθήκευση της ακριβούς χρονικής στιγμής που πραγματοποιήθηκε ένα γεγονός. Αντίστοιχα η δεύτερη λειτουργία διαθέτει την δυνατότητα μεταβολής της στάθμης ενός ακροδέκτη, σε περίπτωση που ο χρονιστής/μετρητής φτάσει μια προκαθορισμένη τιμή και ο λόγος είναι για να μπορούμε να προγραμματίζουμε μια ενέργεια σε μια μελλοντική στιγμή.

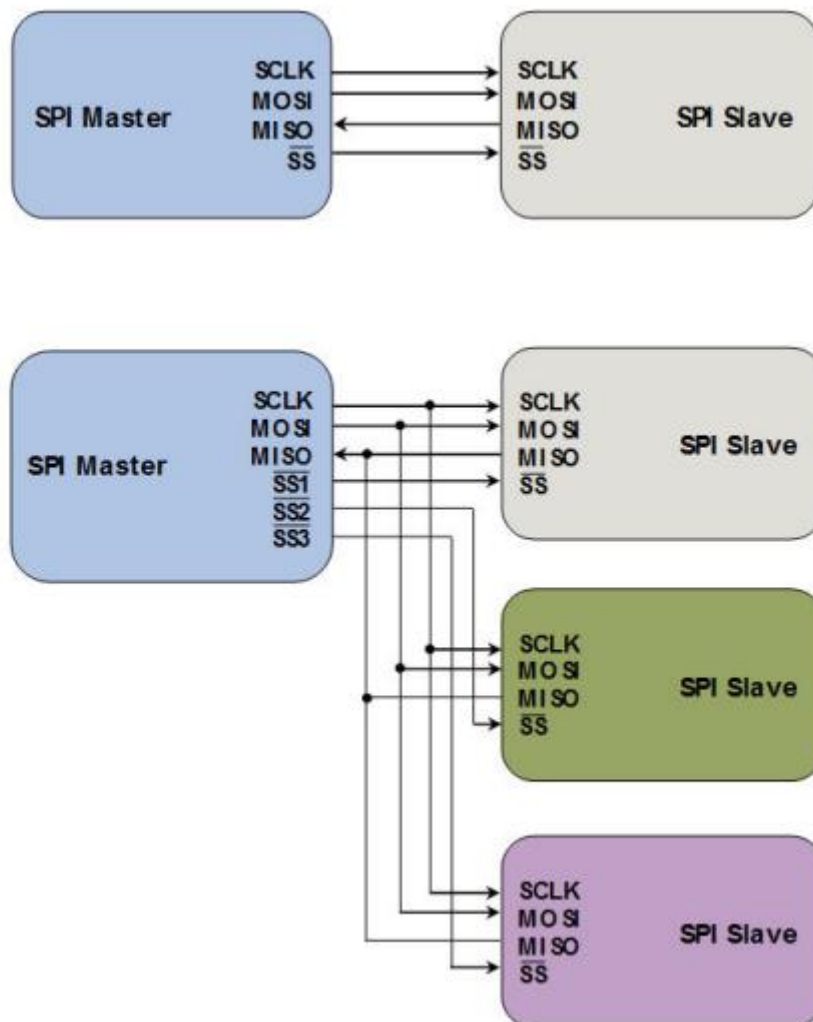
2.4 Περιφερειακά επικοινωνίας

Η επικοινωνία μεταξύ ηλεκτρονικών συστημάτων έχει ένα κοινό χαρακτηριστικό με την ανθρώπινη επικοινωνία και αυτό είναι το γεγονός ότι και οι δύο πλευρές πρέπει να μιλάνε την ίδια γλώσσα. Στον κόσμο των ηλεκτρονικών συστημάτων η γλώσσα μεταφράζεται ως πρωτόκολλο επικοινωνίας. Όσον αφορά το STM32 για την επικοινωνία δύο κυκλωμάτων ή περιφερειακών στην ίδια πλακέτα ανάπτυξης PCB (Printed Circuit Board), προσφέρονται τα πρωτόκολλα SPI (Serial Peripheral Interface) και I2C (Inter Integrated Circuit), ενώ αντίστοιχα για επικοινωνίες μεταξύ ολοκληρωμένου συστήματος και υπολογιστή προσφέρεται το επικοινωνιακό πρωτόκολλο USB και επιπλέον το USART.

2.4.1 Πρωτόκολλο επικοινωνίας SPI

Το SPI είναι ένα πρωτόκολλο επικοινωνίας που χαρακτηρίζεται από την γρήγορη ταχύτητά του, γι αυτό και εφαρμόζεται σε αρκετές σύγχρονες συσκευές ως μέσω επικοινωνίας με άλλα περιφερειακά. Βασίζεται στο μοντέλο σύγχρονης λειτουργίας, πράγμα που καθιστά την παραλαβή και την αποστολή δεδομένων να πραγματοποιείται σύμφωνα με τους παλμούς του ρολογιού, που συγχρονίζει το εκάστοτε σύστημα. Επίσης, διαθέτει δύο βασικές λειτουργίες, με την πρώτη να στηρίζεται στην επικοινωνία ενός αφέντη (master) και ενός σκλάβου (slave) και η δεύτερη σε επικοινωνία ενός αφέντη με πολλούς σκλάβους. Ο STM32 προσφέρει τρία περιφερειακά SPI, με κύριο χαρακτηριστικό την αμφίπλευρη επικοινωνία. Η εμφάνισή

τους στην αρχιτεκτονική του συστήματος παρατηρείται στους διαύλους επικοινωνίας, όπου το ένα από αυτά βρίσκεται στον δίαυλο APB2 με συχνότητα λειτουργίας 72 MHz και τα υπόλοιπα στον δίαυλο επικοινωνίας APB1 με συχνότητα 36 MHz. Αξίζει να σημειωθεί ότι η τα δύο από τα τρία SPI περιφερειακά που βρίσκονται στον δίαυλο APB1 μπορούν να διαμορφωθούν και ως περιφερειακά για μεταφορά δεδομένων ήχου, ορίζοντας το πρωτόκολλο I2S (Inter-Integrated Sound). Συνεχίζοντας, η μεταφορά δεδομένων γίνεται από λέξεις των 8 ή 16 bit με το πρώτο byte να είναι είτε το πιο σημαντικό (MSB – Most Significant Bit), είτε το λιγότερο σημαντικό (LSB- List Significant Bit).

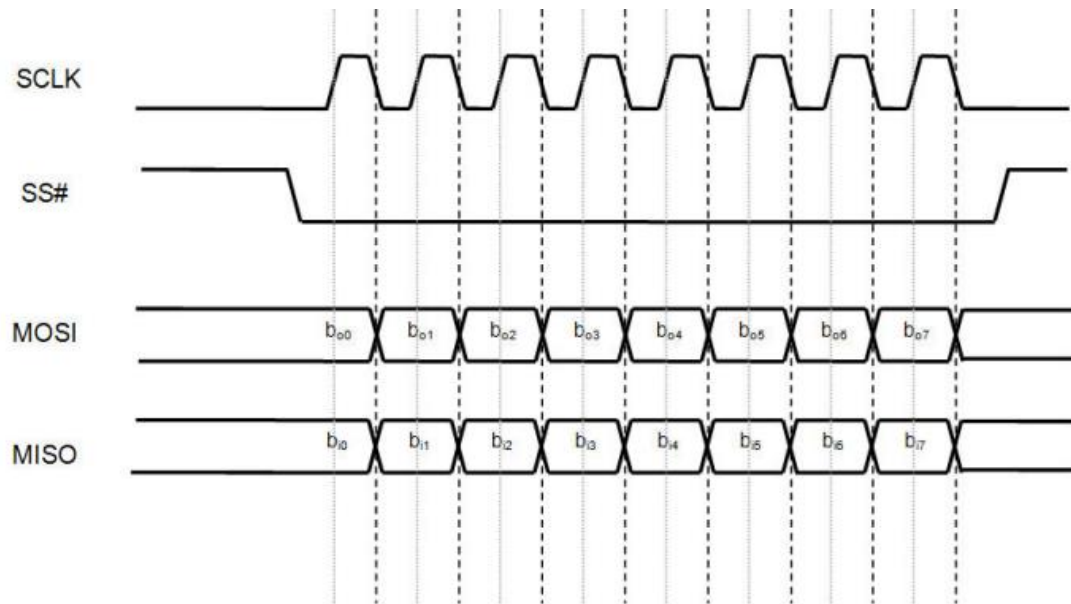


Εικόνα 12 Λειτουργίες μονής και πολλαπλής επικοινωνίας SPI

Η σύνδεση του SPI με άλλα περιφερειακά ή συσκευές βασίζεται σε τέσσερις ακροδέκτες. Οι δύο από αυτούς αφορούν την μεταφορά δεδομένων και χαρακτηρίζονται ως MISO (Master In Slave Out) και MOSI (Master Out Slave In). Το MOSI χρησιμοποιείται όταν ο αφέντης (master) επιθυμεί να διαβάσει δεδομένα από τον σκλάβο (slave), ενώ αντίστοιχα το MISO χρησιμοποιείται όταν ο αφέντης στέλνει δεδομένα προς τον σκλάβο. Ο επόμενος ακροδέκτης ονομάζεται SCLK και είναι υπεύθυνος για τον συγχρονισμό, μέσω παλμών ρολογιού, των δεδομένων που μεταφέρονται μεταξύ διάφορων περιφερειακών. Τέλος, έχουμε τον προαιρετικό ακροδέκτη CE (Chip Enable) ή αλλιώς SS (Slave Select) που χρησιμοποιείται για να σηματοδοτεί την έναρξη ή την λήξη μιας επικοινωνίας καθώς επίσης και για την επιλογή συσκευής ή περιφερειακού που λειτουργεί ως σκλάβος, με σκοπό την επικοινωνία με τον αφέντη. Η τελευταία αναφορά ισχύει μόνο στην περίπτωση που η επικοινωνία είναι μεταξύ ενός αφέντη και πολλών σκλάβων.

2.4.1.1 Εκκίνηση της επικοινωνίας

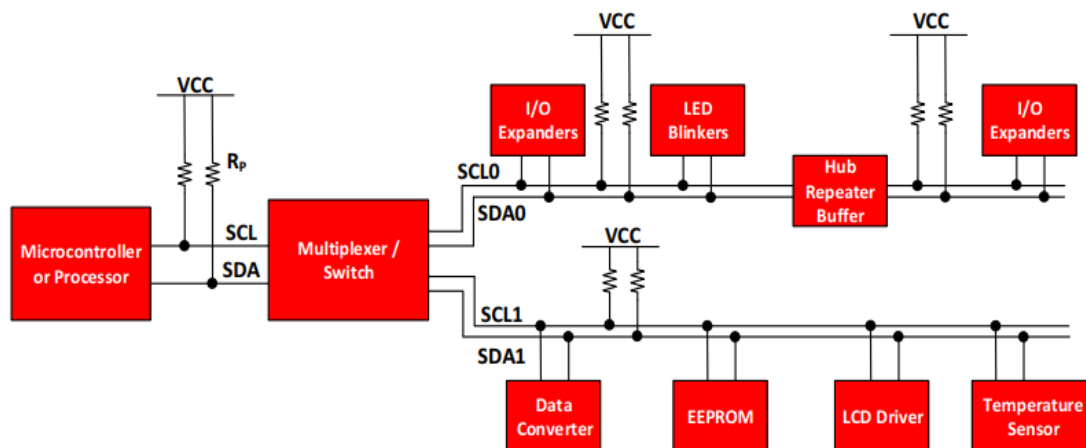
Ο αφέντης (master) είναι υπεύθυνος για την έναρξη της επικοινωνίας με οποιοδήποτε περιφερειακό που ενεργεί ως σκλάβος (slave). Έτσι, όταν έχει δεδομένα προς μετάδοση ή πρέπει να λάβει δεδομένα, τότε επιλέγει την αντίστοιχη συσκευή ή περιφερειακό (σκλάβο) θέτοντας τον ακροδέκτη SS στο λογικό “0” (low). Έπειτα ενεργοποιεί το ρολόι του επεξεργαστή σε μια συμβατή συχνότητα με τον αντίστοιχο σκλάβο. Αν η ενέργεια που θέλει να εκτελέσει ο αφέντης είναι η εγγραφή/μεταφορά δεδομένων τότε προωθεί την εκάστοτε πληροφορία στον ακροδέκτη MOSI και αντίστοιχα αν η ενέργεια απαιτεί λήψη δεδομένων από τον σκλάβο τότε αντλεί την πληροφορία από τον ακροδέκτη MISO. Τέλος, μόλις ολοκληρωθούν οι αντίστοιχες ενέργειες τότε ο αφέντης απενεργοποιεί το SCLK και ελευθερώνει τον σκλάβο θέτοντας τον ακροδέκτη SS στην τιμή λογικό “1”. Ένα αντίστοιχο παράδειγμα διακρίνεται και στην παρακάτω εικόνα, όπου για όση χρονική διάρκεια μεταδίδονται δεδομένα (b0-b7) ο ακροδέκτης SS είναι στο λογικό “0”.



Εικόνα 13 Η λειτουργία του πρωτοκόλλου SPI

2.4.2 Πρωτόκολλο επικοινωνίας I2C

Το πρωτόκολλο I2C είναι ένα αρκετά δημοφιλές μέσω επικοινωνίας που βασίζεται σε ένα σύστημα διαύλων και αφορά λειτουργίες ενός αφέντη και ενός σκλάβου ή πολλαπλών αφεντών (master) και πολλαπλών σκλάβων. Έτσι λοιπόν το I2C αποτελείται από έναν δίαυλο στον οποίο μπορούν να χωρέσουν αρκετά περιφερειακά ή συσκευές, με σκοπό την επίτευξη της επικοινωνίας τους, έχοντας μόνο δύο ακροδέκτες που συνδέονται με τον επεξεργαστή. Είναι ένα πρωτόκολλο που βασίζεται στην αμφίδρομη επικοινωνία θέτοντας την έξοδο του ενός ακροδέκτη του ως open drain, δηλαδή χρησιμοποιώντας ένα καλώδιο να μπορεί η συσκευή που ενεργεί ως αφέντης να στέλνει και να διαβάζει δεδομένα. Ένα παράδειγμα μιας σύνθετης επικοινωνίας στον



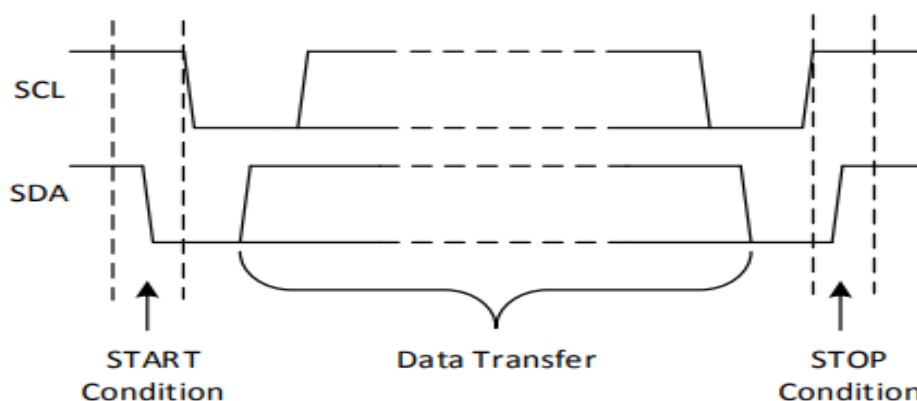
δίαυλο του πρωτοκόλλου απεικονίζεται στην παρακάτω εικόνα.

Εικόνα 14 Δίαυλος επικοινωνίας με δύο ακροδέκτες στο I2C

Ο πρώτος ακροδέκτης είναι το σειριακό ρολόι SCL (Serial Clock Line) και αφορά τον συγχρονισμό μέσα στον δίαυλο του I2C. Ο δεύτερος ακροδέκτης ονομάζεται SDA (Serial Data Line) και είναι υπεύθυνος για την μεταφορά και την λήψη δεδομένων. Στον δίαυλο υποστηρίζονται ταχύτητες από 100kHz για την κύρια και έως 400 kHz για την υψηλότερη. Επιπλέον, προσφέρονται διευθύνσεις από 7 έως 10 bit, ενώ παράλληλα υποστηρίζονται δύο επιλογές ασφάλειας υπό την μορφή διακοπών (interrupts), που η μια προσφέρεται για την ανίχνευση λαθών μέσα στον δίαυλο, ενώ η άλλη για τον έλεγχο των διευθύνσεων και της μετάδοσης δεδομένων.

2.4.2.1 Η διαδικασία επικοινωνίας

Μέσα στον δίαυλο, ενεργός είναι μόνιμα ο αφέντης και κάθε συσκευή που ενεργεί ως slave κατέχει και μια διεύθυνση. Όταν ο αφέντης θέλει να ενεργοποιήσει μια διαδικασία με κάποιον σκλάβο πρώτα στέλνει την διεύθυνσή του και έπειτα ενεργοποιείται η συσκευή για την επικοινωνία. Είναι σημαντικό να αναφερθεί ότι πολλές συσκευές που έχουν τον ρόλο του σκλάβου διαθέτουν καταχωρητές στην μνήμη τους που κατέχουν αρκετές πληροφορίες, πράγμα που σημαίνει ότι για μια τέτοια επικοινωνία ο αφέντης οφείλει να ξέρει τόσο την διεύθυνση του σκλάβου αλλά και την διεύθυνση του καταχωρητή που επιθυμεί να γράψει ή να διαβάσει. Παρ' όλα αυτά, υπάρχουν και συσκευές σκλάβοι που δεν έχουν καταχωρητές και για την επικοινωνία με τον αφέντη χρειάζεται απλά η διεύθυνσή τους. Μια διαδικασία ξεκινά και τελειώνει πάντα από τον αφέντη, έχοντας ως πρώτο βήμα την κατάσταση εκκίνησης ή αλλιώς start condition και ως τελευταίο βήμα την κατάσταση τερματισμού ή αλλιώς stop condition. Πιο συγκεκριμένα, μια μετάβαση από το λογικό "1" στο λογικό "0" του ακροδέκτη SDA, ενώ ταυτόχρονα ο ακροδέκτης SCL είναι στο λογικό "1", σηματοδοτεί την έναρξη της επικοινωνίας και της διαδικασία που έχει ορίσει ο αφέντης. Αντίστοιχα, η μετάβαση από λογικό "0" σε λογικό "1" του ακροδέκτη SDA, όταν το SCL είναι στο λογικό "1", σηματοδοτεί τον τερματισμό της διαδικασίας που ακολούθησε.



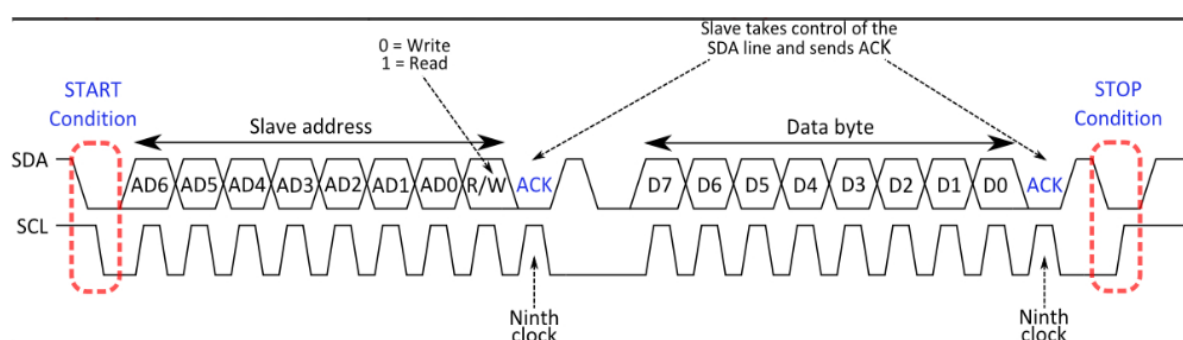
Εικόνα 15 Παράδειγμα Εκκίνησης και τερματισμού μιας διαδικασίας

Συνεχίζοντας, σε κάθε παλμό ρολογιού μεταφέρεται και ένα bit δεδομένων μέσα στον δίαυλο. Τα δεδομένα που μεταφέρονται είναι bytes και κάθε byte έχει οκτώ bits. Ένα byte μπορεί να χαρακτηρίζεται από την διεύθυνση του σκλάβου που επιθυμεί ο αφέντης να επικοινωνήσει ή το μήνυμα που θέλει να στείλει/διαβάσει. Κάθε byte που μεταφέρεται μέσα στον δίαυλο πρέπει να γίνεται ανάμεσα από την κατάσταση εκκίνησης και τερματισμού και ταυτόχρονα να παραμένουν σταθερά στις εναλλαγές που εκτελούνται στον ακροδέκτη SDA όταν γίνεται η αλλαγή της κατάστασης. Μόλις ολοκληρωθεί η μεταφορά ενός byte (διεύθυνση σκλάβου/μήνυμα) ακολουθείται πάντα ένα bit επιβεβαίωσης του παραλήπτη σκλάβου γνωστό ως ACK (Acknowledge), το οποίο σημαίνει ότι κάθε bit που μεταφέρθηκε μέσα στον δίαυλο παραλήφθηκε. Έτσι, ο αφέντης γνωρίζει ότι το μήνυμα του παραδόθηκε και μπορεί έπειτα να στείλει άλλο ή να τερματίσει την επικοινωνία. Αξίζει να σημειωθεί ότι για να μπορέσει ο σκλάβος να στείλει ACK bit, πρέπει ο αφέντης να ελευθερώσει τον ακροδέκτη SDA και ο σκλάβος να τον οδηγήσει στην τιμή λογικό "0". Αν για κάποιο λόγο κατά την διάρκεια της αποστολής του bit επιβεβαίωσης η γραμμή του ακροδέκτη SDA παραμείνει στο λογικό "1", τότε αυτό σηματοδοτεί την αρνητική επιβεβαίωση παραλήπτη γνωστή και ως NACK (Not Acknowledge), πράγμα που σημαίνει ότι το μήνυμα του αποστολέα δεν παραλήφθηκε.

2.4.2.2 Παράδειγμα επικοινωνίας αφέντη σκλάβου

Για την αποστολή δεδομένων μέσα στον δίαυλο του I2C ο αφέντης ξεκινά πάντα την διαδικασία εκκίνησης "start condition" και στην συνέχεια στέλνει την διεύθυνση του παραλήπτη σκλάβου με τελευταίο bit πάντα την ενέργεια εκτέλεσης, δηλαδή είτε εγγραφή είτε ανάγνωση. Το bit αυτό συμβολίζεται ως R/W (Read/Write) όπου αν η τιμή του είναι μηδέν "0" σηματοδοτεί την διαδικασία εγγραφής, ενώ αντίθετα αν είναι ένα "1" σηματοδοτεί την διαδικασία ανάγνωσης δεδομένων από τον σκλάβο. Το

επόμενο βήμα είναι σημαντικό καθώς όταν λάβει ACK bit επιβεβαίωσης (ο παραλήπτης έχει ενεργοποιηθεί και περιμένει την εκτέλεση της διαδικασίας) ο αφέντης, πρέπει σε περίπτωση που ο σκλάβος έχει καταχωρητές να εντοπίσει την εκάστοτε διεύθυνσή του και να στείλει στον δίαυλο μετά το πρώτο ACK και την επόμενη διεύθυνση του καταχωρητή του σκλάβου. Μόλις λάβει και από εκεί bit επιβεβαίωσης συνεχίζει με την ενέργεια που επιθυμεί να εκτελέσει, η οποία καθορίζεται από το bit R/W. Αν ο σκλάβος δεν έχει καταχωρητές, τότε μόλις λάβει το πρώτο ACK ο αφέντης μπορεί να ξεκινήσει την διαδικασία του.



Εικόνα 16 Παράδειγμα επικοινωνίας με συσκευή σκλάβου χωρίς καταχωρητές

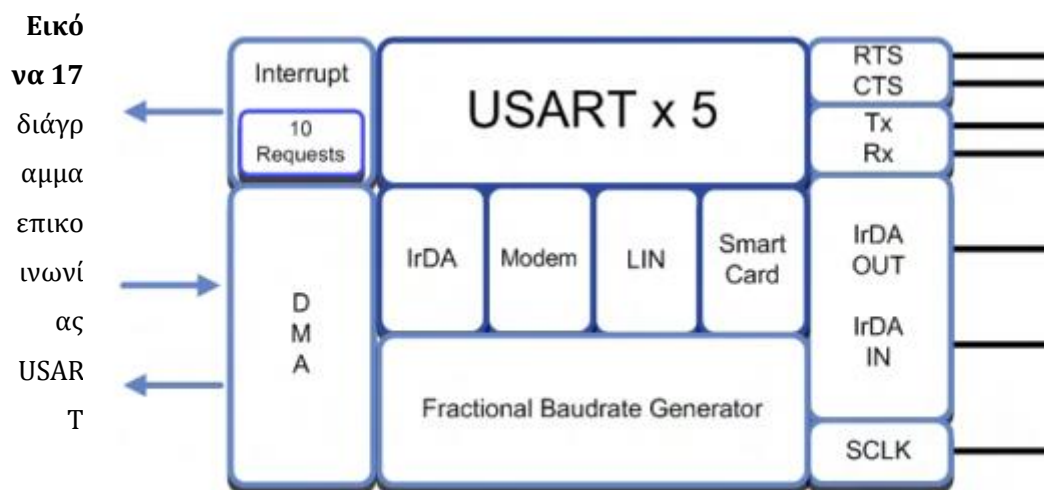
2.4.3 Πρωτόκολλο επικοινωνίας USB

Το USB είναι ένα περιφερειακό επικοινωνίας που συνδέει τον δίαυλο USB 2.0 υψηλής ταχύτητας με το δίαυλο APB. Τα κύρια χαρακτηριστικά του είναι ότι προσφέρει έως οκτώ ακραία σημεία (end points) προς διαμόρφωση, δυνατότητα σύνδεσης και αποσύνδεσης του USB, υποστηριζόμενες μεταφορές ίσων χρόνων, μνήμη SRAM στα 1024 bytes, υποστήριξη προδιαγραφών φόρτισης μπαταρίας και διαχείριση ενέργειας. Ο έλεγχος της διεπαφής γίνεται από ένα συγκεκριμένο USB Host, όπως για παράδειγμα ο ηλεκτρονικός υπολογιστής, με ταχύτητα επικοινωνίας τα 12 Mb/sec. Η μεταφορά δεδομένων γίνεται μέσω της διαθέσιμης μνήμης μεγέθους 1024 bytes όπου εξαρτάται από τα ακραία σημεία (end points) που χρησιμοποιούνται. Η κύρια λειτουργία του με τον host είναι η ανίχνευση πακέτων, η διαχείριση μετάδοσης/λήψης δεδομένων και η

επεξεργασία των πακέτων χειραψίας (handshake). Η μνήμη δεσμεύεται με ένα μπλοκ που σχετίζεται με κάθε ακραίο σημείο και υποδηλώνει την θέση τους στην μνήμη. Η μεταφορά των δεδομένων γίνεται από έναν 16 bit καταχωρητή όπου μετά από κάθε μεταφορά δημιουργείται ένα πακέτο χειραψίας (Handshake) και μια διακοπή από το end point που χρησιμοποιήθηκε. Με αυτόν τον τρόπο ο επεξεργαστής μπορεί εύκολα να εντοπίσει λάθη στην μετάδοση όπως απώλεια bit επιβεβαίωσης ACK, over/underun και άλλα. Τέλος, διατίθεται και η λειτουργία αναστολής SUSPEND mode, με την οποία γίνεται μείωση στην κατανάλωση αποφεύγοντας σπατάλες ενέργειας μέσω της μείωσης του ρολογιού USB, ενώ επίσης υπάρχει και η δυνατότητα πλήρης αδράνειας του. Σε περίπτωση που γίνει ανίχνευση μιας συσκευής στις εισόδους του πρωτοκόλλου τότε προκαλείται μερική επανεκκίνηση, ενώ υπάρχει η δυνατότητα υποστήριξης μιας διακοπής (interrupt) με την οποία μπορεί να προκαλέσει την πλήρη επανεκκίνηση της λειτουργίας του ρολογιού.

2.4.4 Σειριακή επικοινωνία USART

Ο STM32 προσφέρει πέντε σειριακές υποδοχές USART (Universal Synchronous/Asynchronous Receiver/Transmitter) που εντοπίζονται στους διαύλους επικοινωνίας APB. Πιο συγκεκριμένα, η μια USART βρίσκεται στον δίαυλο APB2, με λειτουργία συχνότητας 72 MHz, ενώ οι υπόλοιπες βρίσκονται στον δίαυλο APB1, με συχνότητα λειτουργίας 36 MHz. Κάθε USART διαθέτει πιο βελτιωμένες δυνατότητες, πράγμα που γίνεται αντιληπτό από τον ρυθμό επικοινωνίας που προσφέρεται στα 4.5Mbps. Είναι σημαντικό να αναφερθεί πως σε κάθε διαμόρφωση μιας USART επικοινωνίας πρέπει να ορίζονται και κάποια χαρακτηριστικά όπως το μέγεθος των δεδομένων που μπορεί να είναι είτε 8 bit είτε 9 bit, ο ρυθμός μετάδοσης (Baud Rate) καθώς επίσης και το bit ισοτιμίας, δηλαδή άρτια ή περιττή ισοτιμία.



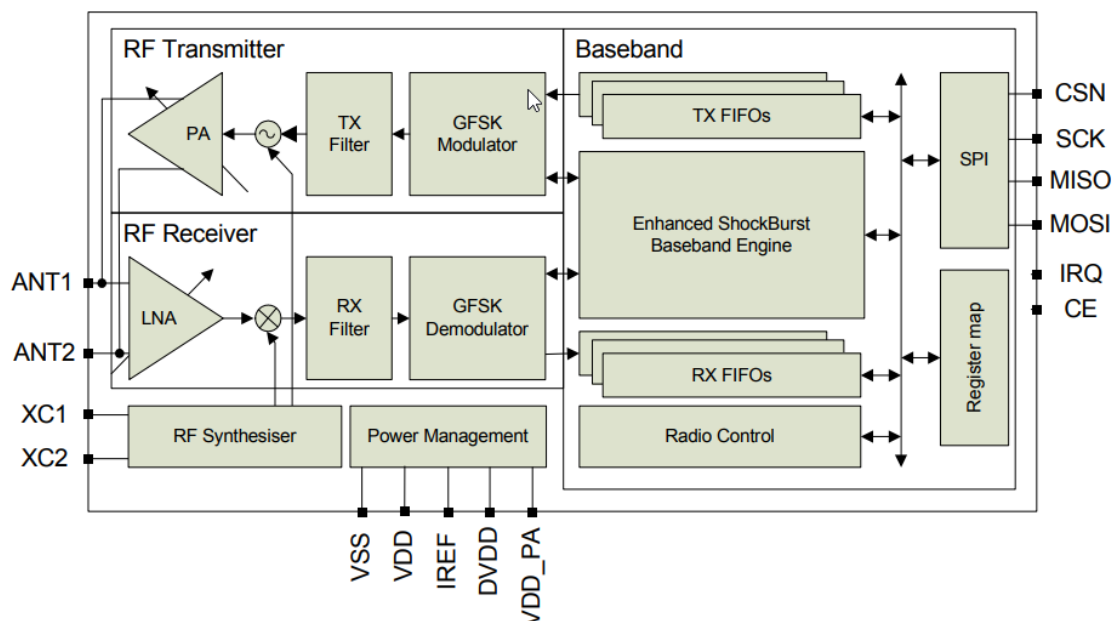
Η λειτουργία του πρωτοκόλλου μπορεί να είναι σύγχρονη ή ασύγχρονη, προσφέροντας επιλογές τόσο ημιαμφίδρομου (half-duplex) όσο και αμφίδρομου (full-duplex) τρόπου επικοινωνίας. Η κύρια λεπτομέρεια του στην επικοινωνία βασίζεται σε δύο κανάλια DMA που ονομάζονται TX (Transmit Data Out) για την αποστολή δεδομένων και RX (Receive Date In) για την ανάγνωση δεδομένων μεταξύ συσκευών. Να σημειωθεί ακόμη πως μπορεί το USART να διαμορφωθεί ως μέσω διασύνδεσης με την μορφή διαύλου για τοπικά δίκτυα Local Interconnection Network – LIN. Επιπλέον, κάθε USART μπορεί να χρησιμοποιηθεί ως σειριακός υπέρυθρος κωδικοποιητής ή αποκωδικοποιητής (SIR encoder/decoder), πράγμα που καθιστά το πρωτόκολλο ικανό για συστήματα επικοινωνιών με υπέρυθρες χρησιμοποιώντας μέγιστο ρυθμό μετάδοσης τα 115200 bps και λειτουργία Return to Zero – NRZ με μικρή κατανάλωση, ρυθμίζοντας το ρολόι από 1.4 MHz έως 2.12 MHz και δημιουργώντας το πρότυπο IrDA (Infrared Data Association). Συνεχίζοντας, το μοντέλο προσφέρει επικοινωνία και με έξυπνη κάρτα (smart card) βασιζόμενο στις προδιαγραφές του προτύπου ISO 7618-3. Η σύγχρονη λειτουργία του, επιτρέπει επικοινωνία και με άλλα περιφερειακά όπως είναι το SPI, με μόνη πρόσθετη λεπτομέρεια την προσθήκη του ακροδέκτη SCLK. Κύριος στόχος του είναι, οι μεταφορές δεδομένων να γίνονται σύμφωνα με τους παλμούς του ρολογιού. Τέλος, οι ακίδες CTS και RTS προσφέρονται για διασύνδεση συσκευής που χρησιμοποιεί USART με modem.

Κεφάλαιο 3. nRF24L01+ ο Ασύρματος

Πομποδέκτης

Το περιφερειακό NRF24L01+ είναι ένας ασύρματος πομποδέκτης, ο οποίος σχεδιάστηκε από την εταιρία Nordic Semiconductor με σκοπό την επίτευξη επικοινωνιών με χαμηλή ισχύ, λειτουργώντας με ρυθμό συχνοτήτων από 2.400 έως 2.4835 GHz. Κύριο χαρακτηριστικό του είναι το πρωτόκολλο βασικής ζώνης που διαθέτει, γνωστό και ως Enhanced Shockburst, το οποίο λειτουργεί με πακέτα μηνυμάτων υποστηρίζοντας νέες βελτιωμένες τεχνολογίες, όπως για παράδειγμα οι ενσωματωμένες ουρές FIFO που συνεισφέρουν στην ομαλή μεταφορά δεδομένων μεταξύ των συσκευών που επικοινωνούν. Το περιφερειακό για την επικοινωνία του με τον επεξεργαστή του συστήματος, χρειάζεται την βοήθεια της σειριακής επικοινωνίας που παρέχει το πρωτόκολλο SPI (Serial Peripheral Interface), ενώ ταυτόχρονα η διαμόρφωσή της ραδιοκεραίας βασίζεται στο μοντέλο GFSK (Gaussian Frequency-Shift Keying), επιτρέποντας στον προγραμματιστή να τροποποιεί κάποιες παραμέτρους όπως για παράδειγμα την συχνότητα καναλιού, την ισχύ του σήματος εξόδου καθώς επίσης και τον ρυθμό μετάδοσης μέχρι τα 2Mbps. Διαθέτει τέσσερις ακροδέκτες εισόδου, ξεκινώντας με τον Chip Enable (CE), ο οποίος ενεργοποιεί τις λειτουργίες αποστολής (TX) και ανάγνωσης (RX) δεδομένων. Έπειτα, υπάρχει ο ακροδέκτης CSN (Chip Select), ο οποίος επιλέγει το περιφερειακό που θέλει η μια συσκευή να επικοινωνήσει και σηματοδοτεί την έναρξη της επικοινωνίας. Στη συνέχεια, έχουμε τον ακροδέκτη του ρολογιού SCK και το MOSI (Master Out Slave In) από το οποίο μεταφέρονται δεδομένα προς το περιφερειακό που ενεργεί ως σκλάβος. Οι δύο ακροδέκτες εξόδου είναι ο MISO (Master In Slave Out) για την μεταφορά δεδομένων προς τον επεξεργαστή και το pin IRQ με το οποίο δημιουργούνται διακοπές από το σύστημα του περιφερειακού nRF. Είναι σημαντικό να αναφερθεί ότι η μέγιστη

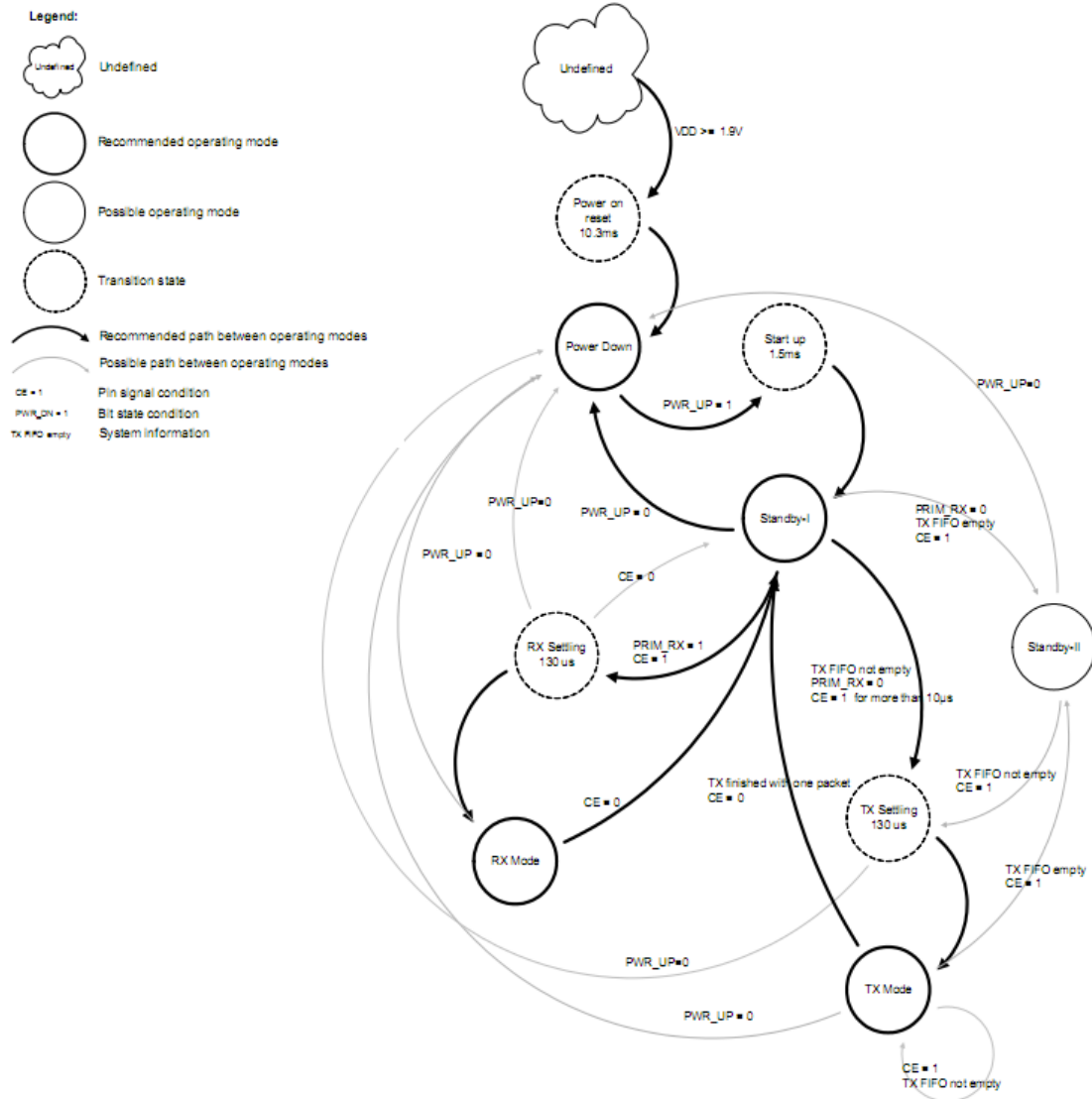
τροφοδοσία του πρέπει να είναι στα 3.6V, ενώ η ελάχιστη στα 1.9V ώστε να μην οδηγηθεί το περιφερειακό σε απροσδιόριστη κατάσταση.



Εικόνα 18 διάγραμμα ασύρματου πομποδέκτη nRF.

3.1 Ραδιοέλεγχος

Το nRF24L01+ έχει ένα ενσωματωμένο μηχανήμα που ελέγχει τις εναλλαγές καταστάσεων του περιφερειακού. Το συγκεκριμένο μηχανήμα διαμορφώνεται από τον χρήστη, ορίζοντας ο ίδιος τιμές στους αντίστοιχους καταχωρητές του. Οι καταστάσεις που μπορούν να λειτουργήσουν στο περιφερειακό nRF24L01+ είναι τέσσερις: η κατάσταση απενεργοποίησης (power down), η κατάσταση αναμονής (standby) και τέλος οι καταστάσεις αποστολής (TX mode) και λήψης (RX mode). Η εικόνα 19 από το εγχειρίδιο χρήσης (user manual) του περιφερειακού, περιγράφει αναλυτικά όσα προαναφέρθηκαν, παρουσιάζοντας τόσο τις απαραίτητες τιμές που πρέπει να έχουν οι καταχωρητές για να οδηγηθούν σε μια κατάσταση, όσο και τις απαραίτητες τιμές τάσης για την λειτουργία του περιφερειακού. Εμβαθύνοντας, όπως παρατηρείται στην παρακάτω εικόνα (Εικόνα 19) όταν η τιμή της τάσης είναι ίση με 1.9V το περιφερειακό ξεκινά να λειτουργεί κατευθυνόμενο στην κατάσταση απενεργοποίησης όπου από εκεί και έπειτα είναι απαραίτητες οι διαμορφώσεις των καταχωρητών του για την εκτέλεση των λειτουργιών του περιφερειακού.



Εικόνα 19 Διάγραμμα καταστάσεων nRF

3.1.1 Power Down- κατάσταση απενεργοποίησης

Το περιφερειακό παραμένει απενεργοποιημένο, ωστόσο η κατανάλωσή του είναι αρκετά μικρή. Η λειτουργία του περιφερειακού επικοινωνίας παραμένει ενεργή όπως και οι αντίστοιχοι καταχωρητές, διατηρούν τις τιμές τους. Η είσοδος στην συγκεκριμένη κατάσταση γίνεται θέτοντας το PWR_UP bit του καταχωρητή CONFIG στην γείωση, δηλαδή τροφοδοτώντας το την τιμή λογικό "0". Αν χρησιμοποιηθεί εξωτερικός ταλαντωτής στο σύστημα τότε ο χρόνος ενεργοποίησης της κατάστασης είναι τα 150 μ s, αντίθετα με την χρησιμοποίηση εξωτερικού κρυστάλλου έχουμε τους χρόνους 1.5 μ s αν $L_s < 30$ mH, 3 μ s αν $L_s = 60$ mH και 4.5 μ s αν $L_s = 90$ mH.

3.1.2 Standby- κατάσταση αναμονής

Η κατάσταση αναμονής είναι μια φάση που περιέχει δύο υποκατηγορίες, εκείνη της κατάστασης αναμονής I (Standby-I) και εκείνη της κατάστασης αναμονής II (Standby-II). Η πρώτη φάση ενεργοποιείται όταν το bit PWR_UP του καταχωρητή CONFIG τεθεί στο λογικό "1", ενώ ταυτόχρονα το περιφερειακό βρίσκεται στην κατάσταση απενεργοποίησης. Βρισκόμενο στην κατάσταση αναμονής 1 (Standby-I), το περιφερειακό διατηρεί σε χαμηλά ποσοστά τόσο την κατανάλωση του, όσο και τον χρόνο εκκίνησης. Σημαντικό στην φάση αυτή είναι ο ακροδέκτης CE του ασύρματου πομποδέκτη καθώς όταν είναι απενεργοποιημένος, δηλαδή τεθεί στο λογικό "0", τότε το περιφερειακό έχει την δυνατότητα επιστροφής στην κατάσταση αναμονής-1 (Standby-I), όταν βρίσκεται σε μια από τις δύο λειτουργίες λήψης (RX) ή αποστολής (TX). Ααντίστοιχα όταν ενεργοποιείται σε αυτή την φάση ξεκινούν οι διαδικασίες δρομολόγησης του περιφερειακού σε μια από τις δύο κατάστασης. Στη δεύτερη φάση, δηλαδή στην κατάσταση αναμονής 2 (Standby-II), μπορεί να βρεθεί το περιφερειακό μόνο από την κατάσταση αποστολής (TX) και έχοντας ενεργοποιημένο τον ακροδέκτη CE. Στην φάση αυτή η κατανάλωση είναι αρκετά υψηλή, καθώς ενεργοποιούνται περισσότεροι ταλαντωτές. Στην περίπτωση που ένα νέο πακέτο έρθει προς μετάδοση στην ουρά εκπομπής πακέτων (TX FIFO), τότε μετά από την μικρή καθυστέρηση του PLL περίπου 130 μ s, ξεκινά η μετάδοση του.

3.1.3 Receiver- κατάσταση λήψης

Η είσοδος του περιφερειακού σε αυτή την κατάσταση γίνεται έχοντας θέση στο λογικό "1" τα bit PWR_UP, PRIM_RX καθώς επίσης και τον ακροδέκτη CE του nRF. Η κατάσταση λήψης είναι μια ενεργή κατάσταση στην οποία οδηγείται το περιφερειακό με σκοπό την λήψη δεδομένων από άλλη συσκευή. Η κύρια λειτουργία του βασίζεται

στην αποδιαμόρφωση των σημάτων καναλιού, ενώ ταυτόχρονα ενεργοποιείται το πρωτόκολλο βασικής ζώνης με σκοπό την αναζήτηση έγκυρων πακέτων, όπου ένα έγκυρο πακέτο χαρακτηρίζεται από μια αξιόπιστη διεύθυνση και αντίστοιχα ένα αξιόπιστο CRC. Σε περίπτωση που ένα έγκυρο πακέτο βρεθεί από το πρωτόκολλο βασικής ζώνης, τότε ελέγχεται η ουρά λήψης πακέτων (RX FIFO), η οποία αν δεν είναι γεμάτη τότε τοποθετείται το πακέτο σε μια κενή θέση μέσα στην ουρά. Απεναντίας, αν είναι γεμάτη η ουρά λήψης τότε το πακέτο απορρίπτεται.

3.1.4 Transmitter- κατάσταση μετάδοσης

Η κατάσταση μετάδοσης είναι μια ενεργή κατάσταση, στην οποία το περιφερειακό παραμένει έως ότου έχει δεδομένα προς μετάδοση στην ουρά εκπομπής πακέτων (TX FIFO). Η είσοδος σε αυτήν την κατάσταση γίνεται, έχοντας το PWR_UP bit στο λογικό “1”, ενώ αντίθετα το PRIM_RX bit στο λογικό “0”. Ταυτόχρονα με αυτές τις διαμορφώσεις των καταχωρητών απαιτείται και ένας παλμός στον ακροδέκτη CE για διάρκεια ίση με 10μs. Μετά την ολοκλήρωση της εκπομπής ενός πακέτου αν ο ακροδέκτης CE μηδενιστεί (CE=0), τότε το περιφερειακό επιστρέφει στην κατάσταση αναμονής I (Standby-I). Αντίθετα, αν το CE παραμένει ενεργό (CE=1) τότε απαιτείται ο έλεγχος της ουράς εκπομπής, όπου αν υπάρχουν ακόμη δεδομένα προς μετάδοση τότε εκτελείται επιπλέον ενέργεια μετάδοσης ωφέλιμου φορτίου. Αν δεν υπάρχουν τότε το περιφερειακό οδηγείται προσωρινά στην κατάσταση αναμονής II (Standby-II). Σε αυτή την κατάσταση για να μπορέσει το περιφερειακό να στείλει δεδομένα ξανά χωρίς να οδηγηθεί στην κατάσταση αναμονής I, τότε πρέπει ο ακροδέκτης CSN του περιφερειακού να είναι ενεργός, δηλαδή να τεθεί στο λογικό “1”, μετά από την εισαγωγή ωφέλιμου φορτίου στην ουρά εκπομπής (TX FIFO).

3.2 Έλεγχος δεδομένων και περιφερειακού

Η διεπαφή ελέγχου δεδομένων δίνει πρόσβαση στις λειτουργίες του περιφερειακού (nRF24L01+), μέσω ειδικών σειριακών εντολών (SPI Commands). Τα δεδομένα, αλλά και τα σήματα εντολών, μεταφέρονται μέσω ακροδεκτών όπου το σύνολό τους είναι έξι. Τέσσερα από αυτά αφορούν την διεπαφή SPI (CSN, SCK, MOSI, MISO) και τα υπόλοιπα δύο είναι οι ακροδέκτες CE [χρησιμοποιείται για την ενεργοποίηση της κατάστασης λήψης (RX)/αποστολής(TX)] και IRQ (χρησιμοποιείται για την διαχείριση των διακοπών). Χρησιμοποιώντας ένα byte από τις εντολές διεπαφής SPI είναι πιο εύκολη η διαχείριση των δεδομένων, όπως για παράδειγμα η πρόσβαση σε μια ουρά εκπομπής (TX_FIFO)/λήψης (RX_FIFO) για διάβασμα ή γράψιμο

δεδομένων αλλά και για την πρόσβαση στο μεγάλο σύνολο καταχωρητών του περιφερειακού. Ο ρυθμός δεδομένων που μπορεί να υποστηριχθεί από την συγκεκριμένη διεπαφή είναι από 0 έως 10 Mbps καθώς επίσης κάθε εντολή SPI αποτελείται από 8 bit.

3.2.1 Εντολές διεπαφής SPI

Κάθε νέα εντολή διεπαφής πρέπει να ξεκινά με μια μετάβαση από το λογικό “1” στο λογικό “0” του ακροδέκτη CSN. Όταν μετατοπίζεται μια εντολή διεπαφής στον ακροδέκτη MOSI, μετατοπίζεται σειριακά επίσης και ο καταχωρητής STATUS στον ακροδέκτη MISO. Η κύρια μορφή των εντολών SPI έχει την εξής μορφή:

<Εντολή: Από το πιο σημαντικό bit (MSBit) στο λιγότερο σημαντικό bit (LSBit)>

<Byte Δεδομένων: Από το λιγότερο σημαντικό bit (LSBit) στο πιο σημαντικό bit (MSBit)>

Στην παρακάτω εικόνα (Εικόνα 20) παρουσιάζονται όλες οι εντολές διεπαφής που μπορούν να χρησιμοποιηθούν με σκοπό να κάνουν την διαμόρφωση του περιφερειακού πιο εύκολη.

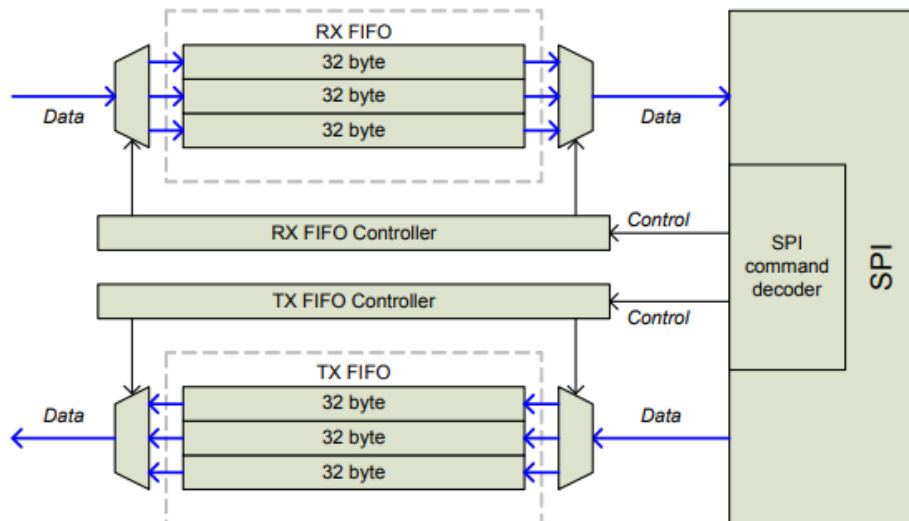
Command name	Command word (binary)	# Data bytes	Operation
R_REGISTER	000A AAAA	1 to 5 LSByte first	Read command and <i>status</i> registers. AAAAA = 5 bit Register Map Address
W_REGISTER	001A AAAA	1 to 5 LSByte first	Write command and status registers. AAAAA = 5 bit Register Map Address Executable in power down or standby modes only.
R_RX_PAYLOAD	0110 0001	1 to 32 LSByte first	Read RX-payload: 1 – 32 bytes. A read operation always starts at byte 0. Payload is deleted from FIFO after it is read. Used in RX mode.
W_TX_PAYLOAD	1010 0000	1 to 32 LSByte first	Write TX-payload: 1 – 32 bytes. A write operation always starts at byte 0 used in TX payload.
FLUSH_TX	1110 0001	0	Flush TX FIFO, used in TX mode
FLUSH_RX	1110 0010	0	Flush RX FIFO, used in RX mode Should not be executed during transmission of acknowledge, that is, acknowledge package will not be completed.
REUSE_TX_PL	1110 0011	0	Used for a PTX device Reuse last transmitted payload. TX payload reuse is active until W_TX_PAYLOAD or FLUSH TX is executed. TX payload reuse must not be activated or deactivated during package transmission.
R_RX_PL_WID ^a	0110 0000	1	Read RX payload width for the top R_RX_PAYLOAD in the RX FIFO. Note: Flush RX FIFO if the read value is larger than 32 bytes.
W_ACK_PAYLOAD ^a	1010 1PPP	1 to 32 LSByte first	Used in RX mode. Write Payload to be transmitted together with ACK packet on PIPE PPP. (PPP valid in the range from 000 to 101). Maximum three ACK packet payloads can be pending. Payloads with same PPP are handled using first in - first out principle. Write payload: 1– 32 bytes. A write operation always starts at byte 0.
W_TX_PAYLOAD_NO_ACK ^a	1011 0000	1 to 32 LSByte first	Used in TX mode. Disables AUTOACK on this specific packet.
NOP	1111 1111	0	No Operation. Might be used to read the STATUS register

Εικόνα 20 Σύνολο εντολών διεπαφής SPI Commands.

3.3 Ουρές δεδομένων

Το nRF24L01+ για την διαχείριση των δεδομένων του στην αποστολή και στην λήψη, χρησιμοποιεί δύο ουρές των 32 byte που ακολουθούν την πολιτική FIFO (First-In-First-Out). Η ουρά αποστολής TX_FIFO αποθηκεύει τα ωφέλιμα φορτία που είναι προς μετάδοση, ενώ η ουρά λήψης RX_FIFO τα ωφέλιμα φορτία που είναι έτοιμα προς διάβασμα. Τόσο η TX_FIFO όσο και η RX_FIFO είναι ορατές και διαθέσιμες προς χρήση

και στις δύο λειτουργίες του περιφερειακού αποστολής-λήψης. Για παράδειγμα, ο πρωτεύων δέκτης PRX χρησιμοποιεί την RX_FIFO για να εξάγει τα πακέτα που μεταδόθηκαν ώστε να ξεκινήσει το διάβασμα των ωφέλιμων φορτίων, αλλά και την TX_FIFO για να αποθηκεύει τα ωφέλιμα φορτία επιβεβαιώσεων ACK, που είναι προς μετάδοση στον PTX. Αντίστοιχα και ο πρωτεύων αποστολέας PTX χρησιμοποιεί την ουρά RX_FIFO για να εξάγει τα πακέτα που μεταφέρουν την επιβεβαίωση ACK και την TX_FIFO, με σκοπό να αποθηκεύει εκεί τα ωφέλιμα φορτία που είναι προς μετάδοση στον πρωτεύων δέκτη PRX. Κάθε ουρά δεδομένων έχει την δυνατότητα να αποθηκεύει μέχρι τρία πακέτα ωφέλιμων φορτίων, ενώ η διαχείρισή τόσο της TX_FIFO όσο και της RX_FIFO γίνεται μέσω των ειδικών εντολών της σειριακής διεπαφής (SPI Commands). Η πλευρά του πρωτεύων αποστολέα για να χρησιμοποιήσει την ουρά αποστολής δεδομένων TX_FIFO, με σκοπό την εγγραφή δεδομένων, χρησιμοποιεί τις εντολές W_TX_PAYLOAD, W_TX_PAYLOAD_NO_ACK και η πλευρά του πρωτεύων δέκτη την εντολή W_ACK_PAYLOAD. Αντίστοιχα για την διαχείριση της ουράς λήψης δεδομένων χρησιμοποιούνται οι εντολές R_RX_PAYLOAD και για τις δύο καταστάσεις λειτουργίας PTX-PRX. Αξιοσημείωτη είναι επίσης και η εντολή FLUSH_TX, η οποία χρησιμοποιείται στην ουρά αποστολής TX_FIFO, με σκοπό να απελευθερώσει τον χώρο της στην περίπτωση που χαθεί η διεύθυνση κάποιου πομπού και κολλήσει η λειτουργία της αποστολής. Όσον αφορά την περίπτωση που ο πρωτεύων αποστολέας επαναλαμβάνει την αποστολή του πακέτου, για τον λόγο του ότι δεν έλαβε επιβεβαίωση, το ωφέλιμο φορτίο δεν φεύγει από την ουρά αποστολής καθ' όλη την διάρκεια προσπαθειών μέχρι να εμφανιστεί η σημαία MAX_RT που σηματοδοτεί την λήξη των προσπαθειών καθώς οι επαναλήψεις έφτασαν τον μέγιστο αριθμό που τους ανατέθηκε. Τέλος, σε κάθε χρονική στιγμή μπορούμε να διαβάσουμε την κατάσταση κάθε ουράς, αν για παράδειγμα είναι άδεια ή γεμάτη μέσω του καταχωρητή STATUS_REGISTER.



Εικόνα 21 Διαχείριση δεδομένων στις ουρές TX_FIFO και RX_FIFO

3.4 Φυσικό κανάλι

Στο περιφερειακό nRF24L01+ μπορεί κανείς να διαμορφώσει τις παραμέτρους του φυσικού καναλιού όπου μεταδίδονται τα δεδομένα. Οι ρυθμίσεις που μπορούν να πραγματοποιηθούν αφορούν, τον ρυθμό μετάδοσης των δεδομένων στον αέρα, την συχνότητα του καναλιού για την σωστή επικοινωνία των συσκευών και την αποφυγή συγκρούσεων και τέλος την διαμόρφωση του ανιχνευτή ισχύος και κατ' επέκταση του ενισχυτή ισχύος.

3.4.1 Ρυθμός μετάδοσης δεδομένων στον αέρα

Ο ρυθμός μετάδοσης δεδομένων στον αέρα είναι ένα σήμα το οποίο διαμορφώνεται μέσω του RF_DR bit στον καταχωρητή RF_SETUP. Σκοπός του είναι η αποστολή και η λήψη δεδομένων μεταξύ των κεραιών δύο ή περισσότερων περιφερειακών (nRF24L01+). Υπάρχουν τρεις διαφορετικές τιμές με τις οποίες μπορούμε να διαμορφώσουμε την μετάδοση δεδομένων στον αέρα και αυτές είναι είτε 250 kbps, είτε 1Mbps, είτε 2Mbps. Όσο πιο χαμηλή είναι η τιμή του ρυθμού μετάδοσης τόσο καλύτερη ευαισθησία υπάρχει στην λήψη των δεδομένων. Ωστόσο με μεγαλύτερες τιμές ρυθμού μετάδοσης μειώνονται κατά πολύ οι πιθανές συγκρούσεις στο ασύρματο κανάλι μετάδοσης. Απαραίτητη προϋπόθεση για την επίτευξη της επικοινωνίας είναι ο ορισμός του ίδιου ρυθμού μετάδοσης στον πομπό και στον δέκτη.

3.4.2 Ραδιοσυχνότητα καναλιού RF

Το εύρος ζώνης του καναλιού ποικίλει ανάλογα με τον ρυθμό δεδομένων που έχει τεθεί για την μετάδοση δεδομένων στον αέρα. Πιο αναλυτικά για ρυθμό μετάδοσης από 250kbps έως 1Mbps το εύρος ζώνης είναι 1 MHz, ενώ αντίθετα για ρυθμό μετάδοσης 2 Mbps είναι 2MHz. Η ακρίβεια επιλογής για το κανάλι τίθεται στα 1MHz και παράλληλα το περιφερειακό λειτουργεί σε συχνότητες από 2,400GHz έως 2,525GHz. Για τις συχνότητες 1Mbps ή 250kbps δεσμεύεται το ίδιο ή λιγότερο εύρος ζώνης στο κανάλι, ενώ για μεγαλύτερες συχνότητες δεσμεύεται περισσότερο εύρος ζώνης στο κανάλι από την ακρίβεια επιλογής πράγμα που μπορεί να οδηγήσει στην επικάλυψη κάποιων καναλιών. Ωστόσο, για να αποφευχθεί αυτό το φαινόμενο στις μεγαλύτερες συχνότητες ρυθμού μετάδοσης πρέπει η απόσταση μεταξύ των καναλιών να είναι μεγαλύτερη των 2MHz. Αξίζει να σημειωθεί ότι τόσο ο πομπός όσο και ο δέκτης οφείλουν να έχουν την ίδια συχνότητα καναλιού για την επίτευξη της επικοινωνίας, η οποία ρυθμίζεται από τον καταχωρητή RF_CH με βάση την παρακάτω φόρμουλα:

$$F_0 = 2400 + \text{RF_CH [MHz]}.$$

3.4.3 Ανιχνευτής ισχύος RPD

Ο ανιχνευτής ισχύος μπορεί να ενεργοποιηθεί όταν το περιφερειακό βρίσκεται στην κατάσταση λήψης RX_MODE. Κύριο χαρακτηριστικό του είναι η στιγμιαία ενημέρωση για το επίπεδο λαμβανόμενης ισχύος στο κανάλι. Ο RPD, εντοπίζεται στον καταχωρητή 09 στο πρώτο του bit και η ενεργοποίησή του βασίζεται σε κάποιες τιμές λογαριθμικής μονάδας ισχύος. Παράδειγμα αν η λαμβανόμενη ισχύς στο κανάλι είναι λιγότερη από τα -64dBm τότε ο ανιχνευτής ισχύος είναι απενεργοποιημένος (RPD=0), ενώ αν είναι μεγαλύτερη από τα -64dBm τότε μπορεί να ενεργοποιηθεί.

3.4.3.1 Ενισχυτής ισχύος

Ο ανιχνευτής ισχύος ή αλλιώς PA (Power Amplifier) είναι υπεύθυνος για τον καθορισμό της ισχύος εξόδου του περιφερειακού. Ο έλεγχος του PA γίνεται μέσω του καταχωρητή RF_SETUP στα bit's RF_PWR όπου στην κατάσταση λήψης TX_MODE μπορεί να διαμορφωθεί με βάση τεσσάρων επιλογών, όπως διακρίνεται και στην παρακάτω εικόνα (Εικόνα 22) από τις οδηγίες της εταιρίας Nordic Semiconductor.

SPI RF-SETUP (RF_PWR)	RF output power	DC current consumption
11	0dBm	11.3mA
10	-6dBm	9.0mA
01	-12dBm	7.5mA
00	-18dBm	7.0mA

Εικόνα 22 Επιλογές για την ισχύ εξόδου του nRF24L01+

3.5 Πρωτόκολλο Enhanced Shockburst

Σύμφωνα με τις πληροφορίες που αναφέρονται στο βιβλίο προδιαγραφών του περιφερειακού (nRF24L01+) από την εταιρία Nordic Semiconductor, το Enhanced Shockburst είναι ένα πρωτόκολλο που ενσωματώνει βοηθητικές λειτουργίες για το περιφερειακό όπως για παράδειγμα, η αυτόματη μετάδοση πακέτων επιβεβαίωσης (Acknowledgment packets), ο αυτόματος σχηματισμός των πακέτων καθώς επίσης και η δυνατότητα αναμετάδοσης, όταν για κάποιο λόγο χαθεί το ωφέλιμο φορτίο που μεταδίδεται ή ξεπεραστεί ο χρόνος αναμονής που έχει τεθεί για το πακέτο επιβεβαίωσης. Επίσης, προσφέρεται η λειτουργία δυναμικού μήκους για το ωφέλιμο φορτίο από 1 έως 32 bytes, αλλά και η δυνατότητα προσαρμογής του περιφερειακού στο να δέχεται περισσότερες από μια πληροφορίες, γνωστή και ως λειτουργία multi receiver, χρησιμοποιώντας έξι τύπους υποδοχής, σε τοπολογία αστέρα. Το συγκεκριμένο πρωτόκολλο χαρακτηρίζεται από την αμφίδρομη επικοινωνία του, η οποία αποτελείται από δύο πομποδέκτες, όπου ο ένας εκ των δύο λειτουργεί ως πρωτεύων δέκτης (PRX – Primary Receiver) και ο άλλος ως πρωτεύων αποστολέας (PTX- Primary Transmitter). Η διαδικασία της επικοινωνίας καθορίζεται από τον πρωτεύων αποστολέα, καθώς από εκείνον ξεκινά και τελειώνει η μετάδοση ενός ωφέλιμου φορτίου. Πιο αναλυτικά, όταν γίνει αποστολή ενός ωφέλιμου φορτίου, σε μορφή πακέτου, ο πρωτεύων αποστολέας περιμένει μήνυμα επιβεβαίωσης από τον πρωτεύων δέκτη και αυτό γίνεται αφού θέτει τον εαυτό του σε λειτουργία δέκτη. Έτσι, όταν ο πρωτεύων δέκτης, λάβει το πακέτο που μεταδόθηκε, το Enhanced Shockburst, μεταδίδει ένα Acknowledgment (ACK) προς τον αποστολέα με σκοπό να επιβεβαιώσει την λήψη του πακέτου που μετέδωσε. Αν για κάποιο λόγο το πακέτο επιβεβαίωσης δεν ληφθεί σύμφωνα με ένα χρονικό όριο, τότε ξεκινά η διαδικασία αναμετάδοσής του, όπου ο αριθμός αναμεταδόσεων αλλά και το χρονικό όριο αναμονής της λήψης του πακέτου επιβεβαίωσης, καθορίζονται από τον χρήστη.

3.4.1 Πακέτο μετάδοσης πρωτοκόλλου

Όταν γίνεται μετάδοση ενός ωφέλιμου φορτίου από τον πρωτεύων αποστολέα, το πρωτόκολλο επικοινωνίας Enhanced Shockburst, δημιουργεί ένα πακέτο μετάδοσης όπου ενσωματώνει μέσα σε αυτό τόσο το ωφέλιμο φορτίο όσο και ένα προοίμιο. Επιπλέον, δύο πεδία όπου το ένα είναι η διεύθυνση του δέκτη και το άλλο είναι για τον έλεγχο του πακέτου και τέλος το πεδίο κυκλικού ελέγχου πλεονασμού γνωστό και ως CRC (Cyclic Redundancy Check).

Preamble 1 byte	Address 3-5 byte	Packet Control Field 9 bit	Payload 0 - 32 byte	CRC 1-2 byte
-----------------	------------------	----------------------------	---------------------	--------------

Εικόνα 23 Πακέτο μετάδοσης πρωτοκόλλου Enhanced Shockburst

3.4.1.1 Προοίμιο

Το προοίμιο είναι μια ακολουθία από bit και χρησιμοποιείται, από την πλευρά του πρωτεύων δέκτη για την αποδιαμόρφωση της ροής των bit που εμφανίζονται, με σκοπό την ανίχνευση αρκετών μεταβάσεων, ώστε να σταθεροποιηθεί η κατάσταση του δέκτη. Ένα byte από αυτή την ακολουθία μπορεί να έχει δύο μορφές, οι οποίες εξαρτώνται από το πρώτο bit της διεύθυνσης του αποστολέα. Αναλυτικότερα, στην περίπτωση που ξεκινά με 1 έχει την μορφή 10101010, ενώ αν ξεκινά με 0 τότε θα έχει την μορφή 01010101.

3.4.1.2 Διεύθυνση

Το πεδίο διεύθυνσεως καθορίζει την διεύθυνση του δέκτη, με σκοπό να διασφαλιστεί ότι το πακέτο θα ληφθεί από τον σωστό δέκτη, καθώς στην περίπτωση που υπάρχουν αρκετές συσκευές που επικοινωνούν μεταξύ τους, να αποφευχθεί η διασταύρωση πακέτων. Ο προγραμματισμός του πεδίου διεύθυνσης γίνεται μέσω του καταχωρητή SETUP_AW και μπορεί να είναι είτε 3 byte, είτε 4 byte, είτε 5 byte.

3.4.1.3 Έλεγχος πακέτου

Ο έλεγχος του πακέτου καθορίζεται φυσικά από το πεδίο ελέγχου που αποτελείται από συνολικά εννέα bit πακέτου. Πιο αναλυτικά έξι bit καταλαμβάνει ένα πεδίο που ενημερώνει τον δέκτη για το μέγεθος του ωφέλιμου φορτίου σε μορφή byte. Για παράδειγμα, για μηδέν byte έχει την μορφή 000000 και για τριάντα δύο την μορφή 100000. Οτιδήποτε μεγαλύτερο του τριάντα δύο χαρακτηρίζεται ως αδιάφορο. Στην

συνέχεια, δυο bit καταλαμβάνονται από το πεδίο ταυτότητας του πακέτου PID (Packet Identification). Σκοπός του είναι, η ενημέρωση του πρωτεύων δέκτη αν το πακέτο που λαμβάνει είναι από αναμετάδοση ή είναι καινούργιο. Τέλος, 1 bit καταλαμβάνει το πεδίο NO_ACK, που στην ουσία είναι μια σημαία (flag) ώστε να καθορίζει στον πρωτεύων δέκτη αν πρέπει να στείλει πακέτο επιβεβαίωσης ή όχι. Αν για παράδειγμα έχει τιμή 1,

Payload length 6bit	PID 2bit	NO_ACK 1bit
---------------------	----------	-------------

τότε υποδηλώνει ότι δεν χρειάζεται η επιβεβαίωση λήψης του πακέτου.

Εικόνα 24 Πακέτο πεδίου ελέγχου

3.4.1.4 Ωφέλιμο φορτίο

Είναι το πεδίο του πακέτου που αποθηκεύεται το μήνυμα που στέλνει ο πομπός προς τον δέκτη. Το μέγεθος του μπορεί να είναι από 0 έως 32 bytes και να χρησιμοποιηθεί μια από τις δύο εναλλακτικές λύσεις, όσον αναφορά το μήκος του. Να έχει δυναμική, είτε στατική μορφή. Η δυναμική υλοποίηση μήκους δεν απαιτεί τα πακέτα που μεταφέρονται μεταξύ πομπού και δέκτη να έχουν το ίδιο μήκος, σε αντίθεση με την στατική υλοποίηση που είναι απαραίτητο το μήκος των μεταδιδόμενων πακέτων να είναι το ίδιο.

3.4.1.5 Κυκλικός έλεγχος πλεονασμού

Το συγκεκριμένο πεδίο είναι ένας μηχανισμός ανίχνευσης σφαλμάτων, όπου στην περίπτωση που αποτύχει ο έλεγχος κανένα πακέτο δεν μπορεί να μεταδοθεί. Το μέγεθός του μπορεί να είναι ένα ή δύο bytes και ο καθορισμός του βασίζεται σύμφωνα από τους υπολογισμούς των πεδίων διεύθυνσης και ελέγχου. Αν το μέγεθός του είναι ένα byte τότε χρησιμοποιείται το πολυώνυμο $X^8 + X^2 + X + 1$ με αρχική τιμή 0xFF, ενώ αντίθετα αν είναι δύο byte το μέγεθός του τότε χρησιμοποιείται το πολυώνυμο $X^{16} + X^{12} + X^5 + 1$, με αρχική τιμή 0xFFFF.

3.4.2 Αυτόματες συναλλαγές πακέτων

Το πρωτόκολλο επικοινωνίας Enhanced Shockburst, διαθέτει δύο βασικές λειτουργίες αυτόματου χειρισμού όσον αφορά τα μεταδιδόμενα πακέτα. Η πρώτη λειτουργία αφορά τις αυτόματες προσπάθειες αναμετάδοσης ενός πακέτου και η δεύτερη αφορά την αυτόματη αποστολή μηνυμάτων επιβεβαίωσης.

3.4.2.1 Αυτόματη προσπάθεια αναμετάδοσης

Η αυτόματη προσπάθεια αναμετάδοσης είναι μια λειτουργία που σχετίζεται με τον πρωτεύων αποστολέα (PTX). Όταν ξεκινήσει η αποστολή ενός μηνύματος με την μορφή πακέτου ο PTX θέτει τον εαυτό του σε πρωτεύων δέκτη (PRX) με σκοπό να λάβει το μήνυμα επιβεβαίωσης. Στόχος του είναι η αναγνώριση ότι το πακέτο που μεταφέρει το ωφέλιμο φορτίο του προς τον δέκτη λήφθηκε. Έτσι λοιπόν στην περίπτωση που για κάποια χρονική περίοδο αναμονής δεν ληφθεί το πακέτο ο αποστολέας ξαναστέλνει το μήνυμά του. Ο αριθμός των αναμεταδόσεων, αλλά και ο χρόνος αναμονής για την λήψη επιβεβαίωσης, προσδιορίζεται από τον καταχωρητή SETUP_RETR. Κατά την διάρκεια των αναμεταδόσεων όταν ληφθεί το πακέτο επιβεβαίωσης τότε ο πομπός σταματά και προκαλείται μια διακοπή μέσω του ακροδέκτη IRQ που ονομάζεται TX_DS.

3.4.2.2 Αυτόματη αποστολή επιβεβαίωσης

Η αυτόματη αποστολή επιβεβαιώσεων γίνεται από τον πρωτεύων δέκτη (PRX), όταν λάβει ένα πακέτο τότε στέλνεται αυτόματα η επιβεβαίωση προς τον πρωτεύων αποστολέα (PTX). Η ενεργοποίηση του συγκεκριμένου χαρακτηριστικού βασίζεται στην τιμή που θα έχει το πεδίο NO_ACK από το πεδίο ελέγχου, που βρίσκεται στο πακέτο που μεταδίδεται προς τον δέκτη. Βασικό του πλεονέκτημα είναι ότι μειώνεται η μέση κατανάλωση του ρεύματος αλλά και το φορτίο του μικροελεγκτή. Ωστόσο, είναι σημαντικό να αναφερθεί ότι για να χρησιμοποιηθεί η συγκεκριμένη λειτουργία πρέπει να είναι υπό μετάδοση πακέτα με δυναμικό μέγεθος DPL (Dynamic Payload Length). Η αυτόματη αποστολή του ωφέλιμου φορτίου επιβεβαίωσης γίνεται μέσω της εντολής W_ACK_PAYLOAD, η οποία τοποθετεί το ωφέλιμο φορτίο στην ουρά εκπομπής πακέτων (TX_FIFO). Είναι σημαντικό να αναφερθεί ότι το nRF24L01+ μπορεί να θέσει υπό μετάδοση τρία διαφορετικά ωφέλιμα φορτία επιβεβαίωσης στην ουρά εκπομπής πακέτων την ίδια χρονική στιγμή.

3.4.3 Η λειτουργία του πρωτεύων αποστολέα (PTX)

Ένα πακέτο τίθεται υπό μετάδοση όταν το περιφερειακό βρίσκεται στην κατάσταση μετάδοσης. Η ενεργοποίηση αυτής της κατάστασης, όπως έχει προαναφερθεί γίνεται με από τον ακροδέκτη CE του περιφερειακού, όταν τεθεί στο λογικό "1" (CE=1). Αν υπάρχει κάποιο ωφέλιμο φορτίο στην ουρά εκπομπής πακέτων (TX_FIFO), τότε το περιφερειακό ξεκινά την αποστολή του προς τον δέκτη. Σε περίπτωση που, η λειτουργία της αυτόματης αναμετάδοσης πακέτων είναι ενεργοποιημένη, τότε ο πρωτεύων αποστολέας ελέγχει αν η σημαία που βρίσκεται στο πεδίο ελέγχου NO_ACK είναι εξίσου ενεργοποιημένη. Αν είναι (NO_ACK=1), τότε ο PTX

δεν περιμένει κάποιο μήνυμα επιβεβαίωσης, ενώ αντίθετα μπαίνει στην κατάσταση λήψης με σκοπό να λάβει το πακέτο επιβεβαίωσης γνωστό και ως Acknowledge (ACK). Με την λήψη του πακέτου επιβεβαίωσης, ξεκινά ο έλεγχός του. Σε περίπτωση που δεν έχει ωφέλιμο φορτίο και είναι άδειο τότε ενεργοποιείται μόνο η διακοπή TX_DS μέσω του ακροδέκτη IRQ. Ωστόσο, αν το πακέτο επιβεβαίωσης περιέχει κάποιο ωφέλιμο φορτίο, τότε ενεργοποιούνται ταυτόχρονα οι διακοπές συστήματος TX_DS και RX_DR μέσω του ακροδέκτη IRQ πριν το περιφερειακό επιστρέψει στην κατάσταση αναμονής I (STANDBY-I). Αξίζει να σημειωθεί ότι, το περιφερειακό οδηγείται στην κατάσταση αναμονής II (STANDBY-II), όταν το πακέτο επιβεβαίωσης δεν έχει ληφθεί μετά από μια συγκεκριμένη χρονική περίοδο αναμονής. Αν οι προσπάθειες αναμετάδοσης δεν φτάσουν στη μέγιστη τιμή που τους έχει τεθεί τότε το περιφερειακό οδηγείται από την κατάσταση αναμονής II ξανά στην κατάσταση αποστολέα με σκοπό την αναμετάδοση του πακέτου. Αν για κάποιο λόγο δεν γίνει η λήψη του πακέτου επιβεβαίωσης και οι προσπάθειες αναμετάδοσης φτάσουν στην μέγιστη τιμή τους τότε προκαλείται διακοπή στο σύστημα μέσω του ακροδέκτη IRQ που ονομάζεται MAX_RT.

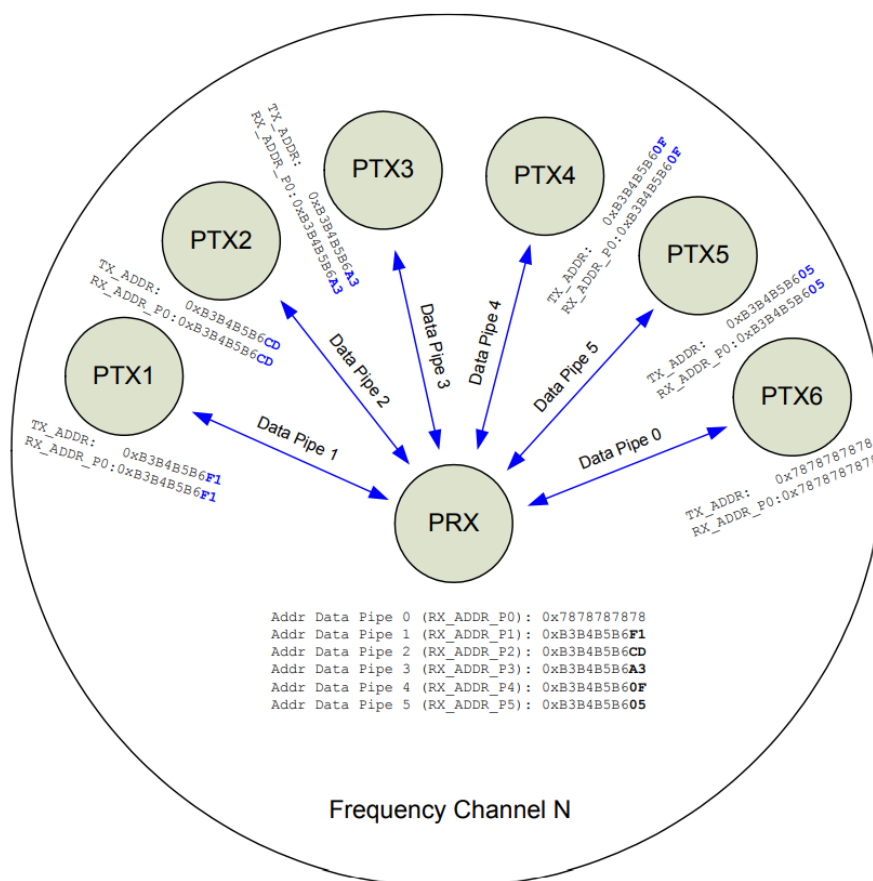
3.4.4 Η λειτουργία του πρωτεύων δέκτη (PRX)

Όταν το περιφερειακό εισέλθει στην κατάσταση λήψης (RX mode), τότε αναζητά συνεχώς πακέτα. Με την έλευση ενός πακέτου γίνεται έλεγχος αν αυτό είναι νέο ή αν είναι αντίγραφο κάποιου που ήδη έχει παραληφθεί και αυτό γιατί αν το πακέτο είναι καινούργιο τότε το ωφέλιμο φορτίο του εισέρχεται κατευθείαν στην ουρά λήψης δεδομένων (RX_FIFO) και ενεργοποιείται η διακοπή RX_DR μέσω του ακροδέκτη IRQ. Ενώ αντίθετα αν στο πακέτο που ληφθεί διαπιστωθεί ότι έχει σταλθεί επιβεβαίωση λήψης (ACK) τότε ενεργοποιείται η διακοπή TX_DS η οποία σηματοδοτεί ότι ο πρωτεύων αποστολέας έλαβε το πακέτο επιβεβαίωσης με ωφέλιμο φορτίο. Ο έλεγχος συνεχίζεται και στο πεδίο του πακέτου που απεικονίζει την σημαία NO_ACK, όπου στην περίπτωση που έχει τιμή ίση με το μηδέν (NO_ACK=0), τότε ο πρωτεύων δέκτης οδηγείται στην κατάσταση αποστολής με σκοπό να στείλει το μήνυμα επιβεβαίωσης και έπειτα οδηγείται ξανά στην κατάσταση λήψης.

3.4.5 Η λειτουργία πολλαπλής λήψης (multireceiver)

Η λειτουργία πολλαπλής λήψης προσφέρει την δυνατότητα της επικοινωνίας ενός nRF24L01+ με άλλα έξι, καθώς υπάρχει ένα σύνολο από έξι διαύλους επικοινωνίας που λειτουργούν παράλληλα. Κάθε δίαυλος έχει μια μοναδική διεύθυνση στο φυσικό κανάλι ραδιοσυχνοτήτων RF, όπου μόνο ένας κάθε φορά μπορεί να μεταφέρει μηνύματα προς τον δέκτη. Σε κάθε δίαυλο μπορεί να εφαρμοστεί το

πρωτόκολλο Enhanced Shockburst καθώς επίσης και να τεθούν σε ισχύ και λειτουργίες όπως η ενεργοποίηση/απενεργοποίηση του CRC, η διαμόρφωση του μήκους των διευθύνσεων στο κανάλι αλλά και του ρυθμού δεδομένων. Αν το πρωτόκολλο επικοινωνίας είναι ενεργοποιημένο τότε πρέπει να οριστεί στον δέκτη η διεύθυνση λήψης του πακέτου, ως διεύθυνση στο πακέτο, για την αποστολή επιβεβαίωσης που πρέπει να στείλει στον πομπό. Η διεύθυνση που ορίζεται σε έναν από τους πέντε καταχωρητές RX_ADDR_Px, όπου x = 0,1,2,3,4,5 πρέπει να είναι μοναδική. Αντίθετα ο πομπός, πρέπει να έχει ίδιες διευθύνσεις στους καταχωρητές TX_ADDR και RX_ADDR_P0 με εκείνες του διαύλου που επικοινωνεί. Αξίζει να σημειωθεί ότι, ο μοναδικός δίαυλος που έχει σταθερή διεύθυνση είναι ο μηδενικός (Data pipe 0), ενώ οι υπόλοιποι διαφέρουν στο τελευταίο byte της διεύθυνσης.

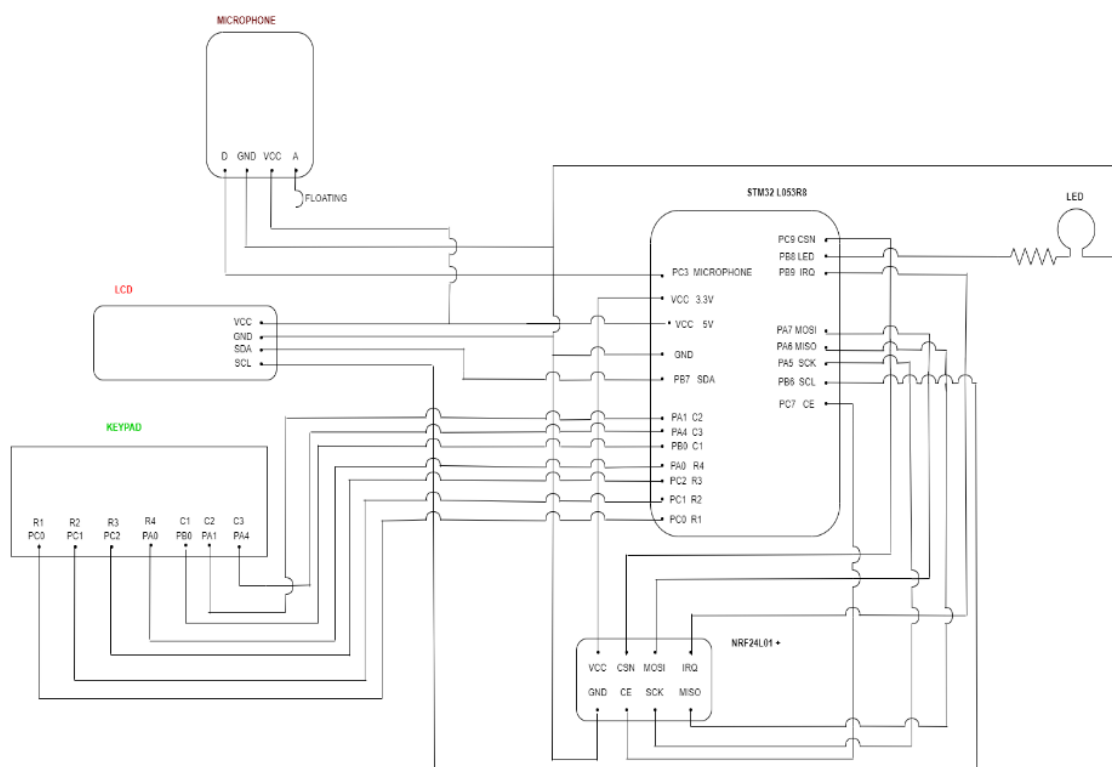


Εικόνα 25 Η λειτουργία των διευθύνσεων στον δακτύλιο της πολλαπλής λήψης

Κεφάλαιο 4. Υλοποίηση

Η υλοποίηση της διπλωματικής εργασίας βασίζεται σε τέσσερα περιφερειακά τα οποία είναι μια οθόνη LCD, ένα πληκτρολόγιο, τον ασύρματο πομποδέκτη nRF24L01+, ένα μικρόφωνο και ένα LED το οποίο κατά την διάρκεια της κανονικής ή αλλιώς normal λειτουργίας του μικροελεγκτή αναβοσβήνει ως ένδειξη σωστής λειτουργίας. Ως γνωστόν ο ασύρματος πομποδέκτης χρησιμοποιείται για την ανταλλαγή μηνυμάτων ασύρματα μέσω της ειδικής κεραίας που διαθέτει το περιφερειακό. Τα μηνύματα αυτά στην λειτουργία του συστήματος που έχει υλοποιηθεί εμφανίζονται στην οθόνη του δέκτη και ο τρόπος αποστολής τους γίνεται μέσω ενός πληκτρολογίου όπου κάθε φορά που ο χρήστης πατά έναν συγκεκριμένο αριθμό, τότε αποστέλλεται ο αριθμός ως μήνυμα στον δέκτη. Στην πλευρά του αποστολέα αν το μήνυμα παραδοθεί σωστά τότε εμφανίζεται ένα μήνυμα επιβεβαίωσης στην οθόνη του που ονομάζεται TX_OK και σηματοδοτεί την σωστή μετάδοση του ωφέλιμου φορτίου. Από την άλλη πλευρά, στον παραλήπτη εμφανίζεται το μήνυμα στην οθόνη με την μορφή RECEIVE: X, όπου X είναι το αντίστοιχο μήνυμα που επέλεξε ο αποστολέας να μεταδώσει. Η υλοποίηση της αποστολής των ωφέλιμων φορτίων γίνεται με την μορφή πακέτου και βασίζεται στο πρωτόκολλο Enhanced Shockburst. Σύμφωνα με όσα προαναφέρθηκαν στο κεφάλαιο 3 όπου και αναλύονται οι πτυχές του πρωτοκόλλου όταν το μήνυμα δεν μεταδοθεί σωστά και φτάσει ο αποστολέας τον μέγιστο αριθμό προσπαθειών εμφανίζεται στην οθόνη του το μήνυμα λάθους MAX_RT (Max Retransmits) πράγμα που ενημερώνει τον χρήστη ότι το πακέτο δεν μπορεί πλέον να μεταδοθεί. Το ίδιο μήνυμα ωστόσο θα εμφανιστεί πάλι στον αποστολέα στην περίπτωση που ξεπεραστεί η μέγιστη απόσταση μεταξύ των δύο ενδιαφερόντων προς επικοινωνία περιφερειακών, η οποία είναι στα 100 μέτρα. Τέλος υπάρχει και από ένα μικρόφωνο τόσο στον πομπό όσο και στον δέκτη που αντιλαμβάνεται τον θόρυβο, και αποστέλλει ο εκάστοτε πρωτεύων πομπός ένα μήνυμα θορύβου το οποίο εμφανίζεται στην οθόνη του δέκτη ως SOUND DETECTED. Στο παρακάτω σχεδιάγραμμα (Εικόνα 26) παρουσιάζεται ο τρόπος με τον οποίο έχουν

συνδεθεί όλα μεταξύ τους και ο τρόπος με τον οποίο επικοινωνούν με τον μικροελεγκτή του συστήματος.



Εικόνα 26 Διάγραμμα υλοποίησης συστήματος.

4.1 Η επικοινωνία του μικροελεγκτή με τον ασύρματο πομποδέκτη - SPI

Για την επίτευξη αυτής της επικοινωνίας χρησιμοποιήθηκε το πρωτόκολλο σειριακής επικοινωνίας SPI (Serial Peripheral Interface). Η υλοποίηση του πρωτοκόλλου βασίστηκε στην λειτουργία των διακοπών (interrupts), δηλαδή κάθε φορά που λαμβάνει ή στέλνει ένα μήνυμα ο πρωτεύων δέκτης/αποστολές παράγεται ένα σήμα διακοπής στην CPU με σκοπό να εκτελεστεί άμεσα το τμήμα κώδικα που ενεργοποιεί το σήμα αυτό. Σύμφωνα με τις οδηγίες που προέρχονται από το εγχειρίδιο χρήσης (data sheet) του μικροελεγκτή για την αρχικοποίηση του πρωτοκόλλου, που γίνεται μέσω της συνάρτησης `spi_init(void)`, αρχικά τίθεται σε λειτουργία το ρολόι του SPI, που ενεργοποιείται μέσω του διαύλου APB. Έπειτα η παρακάτω διαδικασία στην αρχικοποίηση βασίζεται στην διαδικασία της επικοινωνίας μέσα στον διάυλο όπου οι καταχωρητές ολίσθησης στέλνουν τα δεδομένα τους μέσα από τους ακροδέκτες MISO

και MOSI. Η αρχικοποίηση συνεχίζεται, θέτοντας την επιλογή master mode και ταυτόχρονα ενεργοποιείται η επιλογή RX interrupt με σκοπό όταν λάβει κάτι ο πρωτεύων δέκτης να στείλει σήμα διακοπής, αλλά και την επιλογή slave select να γίνεται από τον πηγαίο κώδικα. Τέλος, ενεργοποιείται η συνάρτηση που εκτελείται μόλις σταλθεί ένα σήμα διακοπής στην CPU, η οποία ονομάζεται Interrupt Service Routine (ISR) ή αλλιώς Interrupt Handler.

4.1.1 Η ενεργοποίηση της υπορουτίνας ISR

Η ρουτίνα ISR ενεργοποιείται κάθε φορά που ο master στέλνει ένα μήνυμα προς τον ασύρματο πομποδέκτη και αντίστοιχα κάθε φορά που ο slave δέχεται ένα ωφέλιμο φορτίο. Ο λόγος που γίνεται η ενεργοποίηση της συγκεκριμένης συνάρτησης είναι η εγγραφή και η ανάγνωση των δεδομένων που λήφθηκαν ή στάλθηκαν από και προς έναν συγκεκριμένο προορισμό. Έτσι λοιπόν στην συνάρτηση SPI1_IRQHandler(void), όσο υπάρχουν διαθέσιμα δεδομένα αποθηκεύονται σε μια public μεταβλητή που είναι ένας πίνακας με την ονομασία SPI_RX_BUFFER. Η συγκεκριμένη ενέργεια γίνεται αντιληπτή στον πρώτο σκέλος της συνάρτησης όπου γράφονται δεδομένα στον πίνακα από τον καταχωρητή DR, που είναι υπεύθυνος για την εγγραφή προσωρινών δεδομένων. Πιο κάτω η συνάρτηση διαχειρίζεται δεδομένα προς εγγραφή και η λειτουργία αυτή χαρακτηρίζεται από την αποστολή ενός bit κάθε φορά, συνεχίζοντας μέχρι η μεταβλητή SPI_TX_SIZE γίνει ίση με το μηδέν, πράγμα που σηματοδοτεί την ολοκλήρωση αποστολής του μηνύματος. Έτσι, σύμφωνα με τις οδηγίες του παρέχονται από την οικογένεια της ST-Microelectronics απενεργοποιούμε τον CE ακροδέκτη για την απελευθέρωση του slave (CE=1), καθώς επίσης και το interrupt signal. Όπως αναφέρθηκε και παραπάνω η ενεργοποίηση της ρουτίνας γίνεται και από την συνάρτηση spi_transmit. Έχει δύο ορίσματα, έναν πίνακα που περιέχει το ωφέλιμο φορτίο που επιθυμούμε να στείλουμε προς τον slave και το μέγεθός του. Αρχικά γίνεται αντιγραφή του ωφέλιμου σε μια άλλη μεταβλητή-πίνακα που ονομάζεται SI_RX_BUFFER και έπειτα αρχικοποιούνται κατάλληλες μεταβλητές όπως ένας μετρητής για τα bytes που είτε στάλθηκαν είτε διαβάστηκαν. Στην συνέχεια ενεργοποιείται το SCN pin για να ξεκινήσει η διαδικασία της επικοινωνίας, με ταυτόχρονη αποστολή του πρώτου byte. Για τον λόγο αυτό η μεταβλητή SPI_TX_SIZE είναι ίση με το μέγεθος του μηνύματος -1 καθώς το πρώτο byte όπως αναφέρθηκε και πριν έχει ήδη σταλθεί, άρα το μέγεθος του ωφέλιμου προς μεταφορά μειώνεται κατά ένα. Τέλος, ενεργοποιείται το

TX_INTERRUPT με σκοπό να ξεκινήσει η ρουτίνα για την εγγραφή του ωφέλιμου φορτίου προς τον slave.

4.1.2

4.2 Η εφαρμογή του πρωτοκόλλου Enhanced Shockburst

Για την διαχείριση τόσο των SPI commands όσο και την διαμόρφωση των δεδομένων των καταχωρητών, δημιουργήθηκαν τρεις συναρτήσεις που αφορούν την αποστολή, την λήψη και την εγγραφή μιας εντολής. Οι τρεις συναρτήσεις βασίζονται στην συνάρτηση αποστολής δεδομένων που υλοποιείται στο πρωτόκολλο επικοινωνίας SPI, spi_transmit. Η ρουτίνα που στέλνει την εντολή από το περιφερειακό προς τον μικροελεγκτή έχει ένα όρισμα που είναι η εντολή SPI και η οποία αποστέλλεται κατευθείαν μέσω της συνάρτησης SPI. Οι υπόλοιπες συναρτήσεις χρησιμοποιούν δύο ορίσματα και είναι υπεύθυνες για τον έλεγχο των καταχωρητών του περιφερειακού nRF τόσο για την αρχικοποίηση του με βάση τα πρότυπα του Enhanced Shockburst όσο και για άλλες πιο βασικές λειτουργίες όπως είναι η ενεργοποίηση ειδικών αγωγών για την εναέρια αποστολή μηνυμάτων μεταξύ ενός πρωτεύων δέκτη(PRX) και ενός πρωτεύων αποστολέα(PTX). Όσον αφορά την συνάρτηση εγγραφής δεδομένων, το πρώτο όρισμα είναι ένας καταχωρητής που πρέπει να διαμορφωθεί και το δεύτερο όρισμα είναι τα δεδομένα που ενημερώνουν τον καταχωρητή για την ενεργοποίηση των διαφορετικών πτυχών που μπορεί να προσφέρει. Ένα χαρακτηριστικό παράδειγμα είναι εκείνο που απεικονίζεται στον διάγραμμα καταστάσεων του περιφερειακού nRF24L0+, όπου για να πάμε από την κατάσταση STANDBY-I στην RX_MODE πρέπει το PRIM_RX bit να είναι ενεργοποιημένο, δηλαδή στο λογικό "1". Παρατηρώντας τον καταχωρητή που περιέχει το συγκεκριμένο bit, ο οποίος ονομάζεται CONFIG και έχει reset value την τιμή : 00001000, για την επίτευξη της συγκεκριμένης λειτουργίας αρκεί να χρησιμοποιήσω κατάλληλα την συνάρτηση εγγραφής θέτοντας σωστά δεδομένα στο δεύτερο όρισμα με σκοπό την ανοδική ώθηση του πρώτου bit, στο λογικό "1". Έτσι λοιπόν η ενέργεια αυτή αποτυπώνεται ως εξής:

Nrf_write_command(00000001). Σε περίπτωση που θέλω να ενεργοποιήσω κι άλλες λειτουργίες που μου προσφέρει ο καταχωρητής εφαρμόζω στα κατάλληλα σημεία το λογικό ένα στα δεδομένα του δεύτερου ορίσματος με σκοπό να ενεργοποιήσω τα αντίστοιχα bit. Συνεπώς η spi_transmit στέλνει έναν πίνακα με δύο δεδομένα όπου το πρώτο στο πρώτο όρισμα γίνεται η λογική πράξη OR με εντολή SPI W_REGISTER και αυτό γιατί θέλουμε να «ενημερώσουμε» την λειτουργία που πάμε να κάνουμε στον μικροελεγκτή.

00	CONFIG				Configuration Register
	Reserved	7	0	R/W	Only '0' allowed
	MASK_RX_DR	6	0	R/W	Mask interrupt caused by RX_DR 1: Interrupt not reflected on the IRQ pin 0: Reflect RX_DR as active low interrupt on the IRQ pin
	MASK_TX_DS	5	0	R/W	Mask interrupt caused by TX_DS 1: Interrupt not reflected on the IRQ pin 0: Reflect TX_DS as active low interrupt on the IRQ pin
	MASK_MAX_RT	4	0	R/W	Mask interrupt caused by MAX_RT 1: Interrupt not reflected on the IRQ pin 0: Reflect MAX_RT as active low interrupt on the IRQ pin
	EN_CRC	3	1	R/W	Enable CRC. Forced high if one of the bits in the EN_AA is high
	CRCO	2	0	R/W	CRC encoding scheme '0' - 1 byte '1' - 2 bytes
	PWR_UP	1	0	R/W	1: POWER UP, 0: POWER DOWN
	PRIM_RX	0	0	R/W	RX/TX control 1: PRX, 0: PTX

Εικόνα 27 Ο καταχωρητής CONFIG

4.2.1 Ενεργοποίηση των καταστάσεων TX και RX.

Παρατηρώντας τον αντίστοιχο κώδικα στο <Παράρτημα Α> στην ενότητα <Κατάσταση RX>, αρχικά χρησιμοποιείται μια σημαία (flag) που τίθεται ως barrier όταν γίνεται μεταφορά δεδομένων είτε για την κατάσταση RX είτε για την κατάσταση TX. Από την στιγμή που έχει δοθεί ρεύμα στο nRF βρισκόμαστε στην κατάσταση power down, όπου εκεί μπορούμε να χρησιμοποιήσουμε SPI commands αλλά και γενικότερα SPI λειτουργίες όπως το spi_transmit για παράδειγμα που ενεργοποιείται μέσω της nrf_write_reg. Αρχικά απενεργοποιείται το CE pin και στην συνέχεια πρώτου αλλάζει η κατάσταση προσαρμόζεται ο επιθυμητός αριθμός δεδομένων που πρέπει να σταλούν μέσω του αγωγού που έχει οριστεί. Στην συνέχεια γράφεται στον καταχωρητή CONFIG η τιμή 00001011 μέσω της εντολής **nrf_write_reg(CONFIG,0x0B)** με σκοπό να ενεργοποιηθεί το bit PRIM_RX, το CRC και το PWR_UP. Από την στιγμή που το PWR_UP γίνει “1” θα μεταβεί στην STANDBY-I και έπειτα με το PRIM_RX και το CE pin

θα μετέβει στην RX_MODE κατάσταση όπου εκεί ενεργοποιείται το barrier με σκοπό να ξεκινήσει η διαδικασία της ανάγνωσης του μηνύματος. Αντίστοιχα, η διαδικασία αρχικοποίησης του TX_MODE, ξεκινά με το CE pin απενεργοποιημένο. Στην συνέχεια χρησιμοποιείται η εντολή nrf_write_reg κατάλληλα στον STATUS register για να καθαριστούν όλα τα δεδομένα που σχετίζονται με την TX_FIFO, RX_FIFO αλλά και τις προσπάθειες επαναπροστολής του μηνύματος σε περίπτωση αποτυχίας. Βρισκόμενος στην POWER DOWN κατάσταση μέσω της εντολής nrf_write_payload όπου φορτώνεται το επιθυμητό ωφέλιμο φορτίο που πρέπει να σταλθεί στην TX_FIFO. Έπειτα αλλάζει η κατάσταση καθώς γράφεται στον καταχωρητή η τιμή 00001010 όπου θα ενεργοποιηθεί το CRC, το PWR_UP και θα απενεργοποιηθεί το PRIM_RX bit. Έτσι σύμφωνα με το σχεδιάγραμμα βρισκόμαστε στην TX_MODE κατάσταση όπου εκεί ενεργοποιείται το CE pin και το barrier για να ξεκινήσει η διαδικασία της αποστολής μηνύματος.

4.2.2 Αρχικοποίηση του πρωτοκόλλου.

Για την επικοινωνία χρησιμοποιείται ο αγωγός (pipe) 0. Πριν την αρχικοποίηση του nRF, με βάση το πρωτόκολλο enhanced shockburst, δημιουργούνται δύο πολύ σημαντικές μεταβλητές, όπου η πρώτη αφορά την διεύθυνση του αγωγού και η δεύτερη αφορά την διεύθυνση του αποστολέα. Αυτές οι δύο οι μεταβλητές σύμφωνα με το documentation πρέπει να είναι ίδιες. Στην πρώτη θέση και των δύο πινάκων εκτελείται η λογική πράξη OR μεταξύ του SPI command, που αφορά την εγγραφή των καταχωρητών και τους καταχωρητές RX_ADDR_P0(καθώς χρησιμοποιείται αυτός ο αγωγός και για τα ωφέλιμα φορτία επιβεβαιώσεων ACK, αλλά και για αποστολή μηνύματος) – TX_ADDR. Με την αρχικοποίηση του nRF ενεργοποιείται το CRC καθώς είναι μέρος του πακέτου που στέλνεται από το πρωτόκολλο, το POWER DOWN αλλά και όλα τα interrupts αφού με αυτό τον τρόπο υλοποιείται το συγκεκριμένο σύστημα. Στην συνέχεια τίθεται το data pipe 0 για την μεταφορά των μηνυμάτων ACK και payload, δίνοντας 3 bytes διεύθυνσης. Αφού οριστεί ο μέγιστος αριθμός επαναπροσπαθειών γράφονται οι δύο διευθύνσεις που έχουν ως μεταβλητές nrf_rx_pipe0_address και nrf_tx_address. Τέλος ενεργοποιείται το dynamic payload length, τόσο γενικότερα όσο και στον αγωγό για τις μεταφορές των μηνυμάτων, αλλά και την αυτόματη αποστολή πακέτων/μηνυμάτων επιβεβαίωσης auto ACK.

4.2.3 Ανάγνωση ωφέλιμου φορτίου

Η διαδικασία της ανάγνωσης του μηνύματος πραγματοποιείται από την συνάρτηση nrf_read_payload(void). Από την στιγμή που θα υπάρχει διαθέσιμη πληροφορία δεδομένων για ανάγνωση στο RX_FIFO τότε το περιφερειακό είναι έτοιμο για την ανάγνωση του μηνύματος που λήφθηκε. Για να διαβάσουμε ένα byte σύμφωνα

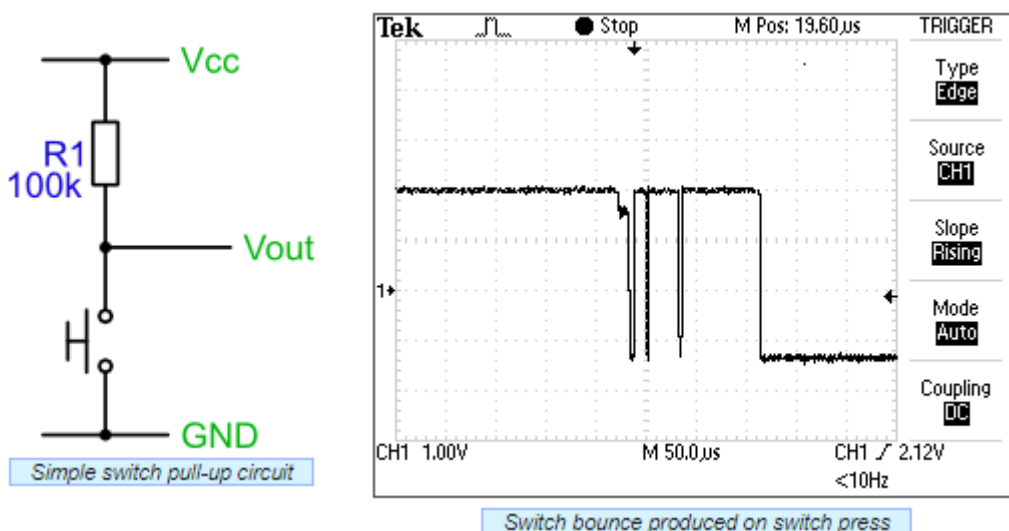
με το SPI πρέπει να στείλουμε ένα dummy byte πρώτα και έπειτα να διαβάσουμε τον DR register όπου αποθηκεύεται το μήνυμα μας. Η διαδικασία ανάγνωσης του DR register πραγματοποιείται από την συνάρτηση SPI1_IRQHandler(void) που ουσιαστικά όταν υπάρχει byte προς ανάγνωση γίνεται interrupt και αποθηκεύει ότι υπάρχει στον DR μέσα στον πίνακα SPI_RX_BUFFER με σκοπό να γίνει η διαχείριση των δεδομένων μετέπειτα. Γυρνώντας λοιπόν στην αρχική συνάρτηση που εκτελεί την ανάγνωση του μηνύματος, αρχικά αποθηκεύεται σε μία μεταβλητή το μέγεθος του μηνύματος που λήφθηκε μέσω της χρήσιμης εντολής R_RX_PL_WID που είναι ένα SPI command. Στην συνέχεια στέλνονται τόσα dummy bytes όσο είναι και το μέγεθος του μηνύματος που λήφθηκε. Έπειτα αρχικοποιείται ένας πίνακας όπου στην πρώτη θέση έχει το SPI command με το οποίο ενημερώνεται το σύστημα για την επιθυμητή λειτουργία, δηλαδή της ανάγνωσης και έπειτα γεμίζονται οι υπόλοιπες θέσεις με dummy bytes. Στην συνέχεια στέλνεται ο πίνακας που δημιουργήθηκε με την βοήθεια της SPI συνάρτησης spi_transmit_wait και ξεκινά η διαδικασία ανάγνωσης του μηνύματος, αποθηκεύοντας το στον πίνακα payload_data. Αναλύοντας την παραπάνω διαδικασία πιο τεχνικά, έστω για παράδειγμα ότι θέλουμε να διαβάσουμε 1 byte, τότε θα στείλουμε πρώτα στο περιφερειακό μέσω SPI συναρτήσεων ένα dummy byte και έπειτα θα διαβάσουμε τον DR register. Από την στιγμή που το slave device βρίσκεται σε transmission mode, το dummy byte που στέλνουμε δεν επηρεάζει τους καταχωρητές αλλά ούτε και τα δεδομένα τους. Όταν θα λάβει ο slave το dummy byte τότε η συσκευή θα στείλει κάνοντας transmit ένα byte. Αυτό θα ενεργοποιήσει το RXNE(Receive Buffer Not Empty) bit και θα σηματοδοτήσει την ύπαρξη νέων δεδομένων στον DR(Data Register) που είναι έτοιμα προς διάβασμα. Η παραπάνω διαδικασία γίνεται με interrupt και υλοποιείται στο SPI1_IRQHandler(void) που όπως γίνεται αντιληπτό μόλις πιάσει σήμα ότι το RXNE είναι ενεργοποιημένο ξεκινά την ανάγνωση του Data Register. Η συνάρτηση που περιγράφηκε είναι διαθέσιμη προς ανάγνωση στο κεφάλαιο <Παράρτημα Α>.

4.3 Switch debouncing

Η συγκεκριμένα έννοια βρίσκει εφαρμογή στην διαχείριση των buttons που διατίθενται στο πληκτρολόγιο, αλλά και του user button με το οποίο ο χρήστης οδηγεί τον μικροελεγκτή σε λειτουργία χαμηλής κατανάλωσης. Όταν πιέζουμε ένα button, δύο μεταλλικά μέρη ενώνονται. Από την πλευρά του χρήστη αυτό φαίνεται πως η επαφή έγινε αμέσως, όμως αυτό δεν είναι απόλυτα σωστό και θα αποδειχθεί στην συνέχεια

από μια απεικόνιση ενός button στον παλμογράφο. Κατά την διάρκεια του πατήματος μέχρι το button να είναι τελείως κλειστό τα μεταλλικά μέρη έρχονται σε επαφή αρκετές φορές σε ένα μικρό διάστημα της τάξης του 1 ms (μικροδευτερολέπτου). Πιο συγκεκριμένα ο διακόπτης ή το button αναπηδά πολλές φορές σε δύο καταστάσεις, εκείνης της επαφής των μετάλλων και εκείνης της μη επαφής. Με λίγα λόγια όταν το μπουτόν είναι τελείως κλειστό, άρα έχει πατηθεί, οι δύο μεταλλικές επαφές διαχωρίζονται και επανασυνδέονται συνήθως 10 έως 100 φορές σε μια περίοδο της τάξης 1ms. Η παραπάνω αναφορά έγινε με σκοπό να τονιστεί το πρόβλημα στην λειτουργία του πατήματος των buttons αν δεν ληφθεί υπόψιν αυτό που προαναφέρθηκε. Συνήθως το hardware λειτουργεί πιο γρήγορα από το λεγόμενο bouncing στο πάτημα του μπουτόν και αυτό έχει ως αποτέλεσμα να νομίζει ότι πατάμε το μπουτόν αρκετές φορές. Έστω για παράδειγμα ότι έχουμε φτιάξει ένα απλό σύστημα που έχει ένα μπουτόν που λειτουργεί σαν μετρητής και κάθε φορά που το πατάμε αυξάνεται η αρχική τιμή που έχουμε ορίσει κατά ένα. Με βάση όσα προαναφέρθηκαν κάνοντας απλά ένα πάτημα τότε ο μικροελεγκτής θα νομίζει ότι πατάμε το μπουτόν αρκετές φορές και θα μας δώσει ένα μη επιθυμητό αποτέλεσμα που θα κυμαίνεται ανάμεσα στις τιμές 10 έως 100 την κάθε φορά, αντί να αυξάνεται κατά ένα.

Στις παρακάτω εικόνες μπορούμε να δούμε και σε μια απλή εφαρμογή όσα αναφέρθηκαν παραπάνω. Αρχικά η πρώτη εικόνα(αριστερά) απεικονίζει ένα μπουτόν όπου το κύκλωμά του έχει μια pull-up αντίσταση και έπειτα η δεξιά εικόνα απεικονίζει την έναρξη του πατήματος του μπουτόν μέχρι να είναι τελείως κλειστό το κύκλωμα και άρα να έχει πατηθεί πλήρως. Εκεί μπορούμε να διακρίνουμε τις σχετικές αναπηδήσεις που γίνονται μέχρι να πατηθεί τελείως το μπουτόν.

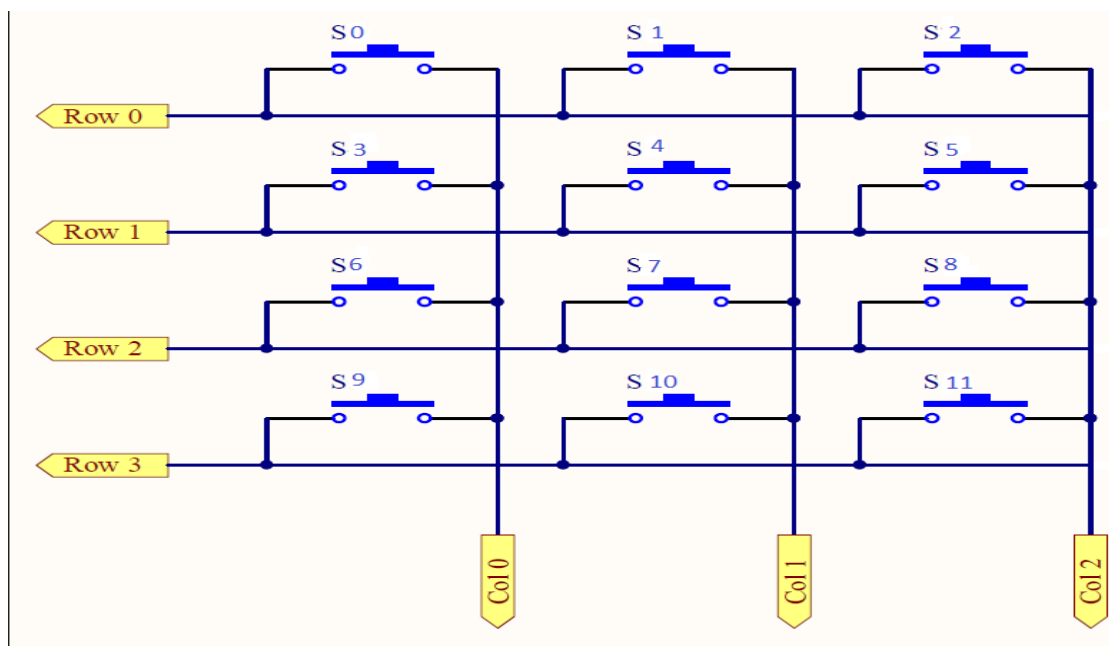


Εικόνα 28 Switch bouncing απεικόνιση σε παλμογράφο

Η επίλυση του παραπάνω προβλήματος γίνεται προγραμματίζοντας κατάλληλα τον μικροελεγκτή ώστε να αγνοεί τις συγκεκριμένες αναπηδήσεις που δημιουργούνται. Η βασική ιδέα είναι η μέτρηση χρονικών διαστημάτων. Έτσι λοιπόν αν το σήμα για ένα συγκεκριμένο χρόνο και παραπάνω είναι σταθερό στο μηδέν τότε το μπουτόν θεωρείται πατημένο. Αρχικά για την μέτρηση των χρονικών διαστημάτων ορίζω έναν μετρητή όπου ελέγχει συνεχώς όλο το σύστημα. Όταν το ενεργοποιηθεί το γεγονός με την μορφή σήματος, που έχει οριστεί μέσω μιας σημαίας, ελέγχεται πρώτα αν ο μετρητής είναι μικρότερος του αριθμού δέκα. Στην περίπτωση που ισχύει αυτό απλά αυξάνεται ο μετρητής και συνεχίζει το πρόγραμμα στην επόμενη δειγματοληψία μέχρι να γίνει η τιμή του μετρητή ίση με τον αριθμό δέκα. Ο λόγος που γίνεται αυτό είναι για να αποφύγουμε τις συνδέσεις και τις αποσυνδέσεις των μετάλλων πριν την σταθεροποίησή τους (κλειστό κύκλωμα). Με τον τρόπο αυτό ο μικροελεγκτής αγνοεί τις μεταπηδήσεις που γίνονται στην αρχή, όπως φαίνεται και στην παραπάνω εικόνα. Όταν ο μετρητής λοιπόν γίνει ίσος με τον αριθμό δέκα τότε ενεργοποιείται η σημαία που σηματοδοτεί το πάτημα του μπουτόν και μηδενίζεται ο μετρητής.

4.3.1 Πληκτρολόγιο

Το πληκτρολόγιο του συστήματος διαθέτει 12 κουμπιά, όπου κάθε κουμπί είναι ένας διακόπτης μεμβράνης. Όλοι οι διακόπτες συνδέονται μεταξύ τους, τόσο στις στήλες όσο και στις γραμμές του πληκτρολογίου μέσα από ένα αγωγίμο ίχνος. Κάθε γραμμή και στήλη στο πληκτρολόγιο έχει και από μια έξοδο που καταλήγει σε έναν ακροδέκτη, όπου το σύνολο όλων των ακροδεκτών είναι 8. Πατώντας ένα κουμπί, λόγω της σύνδεσης μεταξύ τους κλείνει ο διακόπτης του και επιτρέπει την ροή ρεύματος τόσο στην στήλη όσο και στην γραμμή. Για παράδειγμα, βλέποντας στην παρακάτω εικόνα που απεικονίζεται η δομή ενός πληκτρολογίου μεμβράνης, αν πατηθεί το κουμπί S3 τότε βραχυκυκλώνεται η γραμμή 2 με την στήλη 1.



Εικόνα 29 Η εσωτερική δομή του πληκτρολογίου.

Ο τρόπος λειτουργίας του πληκτρολογίου ωστόσο, γίνεται πιο σύνθετος γιατί δεν μπορούμε να γνωρίζουμε πότε ένα συγκεκριμένο κουμπί πατήθηκε καθώς τόσο στην στήλη όσο και στην γραμμή ρέει ρεύμα και άρα όποιος ακροδέκτης έχει τεθεί σε λειτουργία εξόδου θα λαμβάνει θετική ακμή. Η επίλυση του προβλήματος βασίζεται στην λογική της ανίχνευσης σήματος κάθε χρονική στιγμή. Πιο συγκεκριμένα, οι ακροδέκτες που βρίσκονται στις γραμμές τίθενται ως εισόδοι και οι αντίστοιχοι στις στήλες τίθενται ως έξοδοι. Με την βοήθεια ενός χρονιστή/μετρητή, κάθε συγκεκριμένη χρονική στιγμή αλλάζει τις τιμές των εξόδων σε τρεις φάσεις:

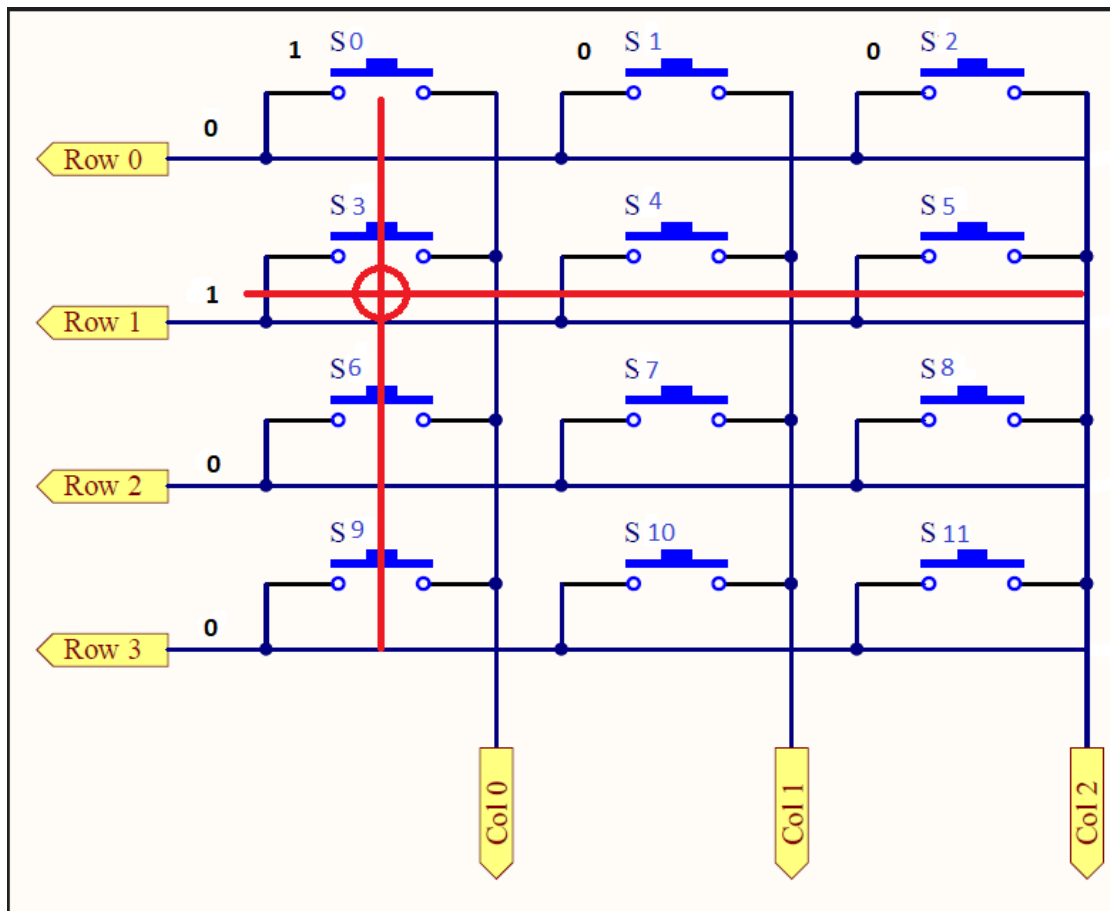
- Φάση 1 : Col1 = 1, Col2 = 0, Col3 = 0
- Φάση 2 : Col1 = 0, Col2 = 1, Col3 = 0
- Φάση 3 : Col1 = 0, Col2 = 0, Col3 = 1

Αντίστοιχα για κάθε χρονική στιγμή οι τιμές των εισόδων στους ακροδέκτες παραμένουν στο λογικό "0", μέχρι κάποιος πατηθεί και ενεργοποιηθεί το σήμα σε λογικό "1". Με τον τρόπο λοιπόν αυτό οποιοδήποτε κουμπί πατηθεί ο μικροελεγκτής μπορεί να καταλάβει ποιο είναι καθώς μόλις αυτό πατηθεί θα ενεργοποιηθεί ο αντίστοιχος ακροδέκτης εισόδου και με την κατάλληλη φάση(1-3) θα γίνει σύνδεση μεταξύ τους όπου το σήμα εξόδου από τον ακροδέκτη που βρίσκεται στην στήλη θα ταυτίζεται με το αντίστοιχο του ακροδέκτη εισόδου και έτσι μπορούμε να γνωρίζουμε πιο κουμπί πατήθηκε. Παράδειγμα, αν πατηθεί το κουμπί S3 τότε οι ακροδέκτες που

βρίσκονται στις γραμμές του πληκτρολογίου και ενεργούν ως είσοδοι θα έχουν τα εξής σήματα στις εισόδους τους:

- Row0 = 0 (Low)
- Row1 = 1 (High)
- Row2 = 0 (Low)
- Row3 = 0 (Low)

Οπότε με την φάση 1 (Col1 = 1, Col2 = 0, Col3 = 0) θα ταυτιστούν οι τιμές των ακροδεκτών και έτσι μπορούμε να ξέρουμε ακριβώς ποιο κουμπί πατήθηκε, όπως φαίνεται και η αποτύπωση του παραδείγματος στην παρακάτω εικόνα. Η διαδικασία που αναλύθηκε προηγουμένως είναι υλοποιημένη από την συνάρτηση `keypad_scanning(uint8_t key)`, που δέχεται ως όρισμα μια είσοδο που αντιστοιχεί σε ένα κουμπί του πληκτρολογίου.



Εικόνα 30 Παράδειγμα αναγνώρισης πατημένου διακόπτη.

4.4 Επικοινωνία μικροελεγκτή με οθόνη LCD - I2C

Για την επικοινωνία του μικροελεγκτή με την οθόνη χρησιμοποιήθηκε το πρωτόκολλο I2C. Ο λόγος που επιλέχθηκε το I2C στο σύστημα είναι για να εμφανίζονται τα μηνύματα που λαμβάνει το περιφερειακό (nRF) στον χρήστη. Ο μικροελεγκτής θα στέλνει μόνο μηνύματα προς την οθόνη, χωρίς να έχει κάποια άλλη απαίτηση, συνεπώς θα ορίζεται ως master και το περιφερειακό (LCD) που είναι ο slave δεν θα στέλνει κάποιο μήνυμα. Σύμφωνα με το datasheet του μικροελεγκτή ως πρώτη επιλογή είναι να επιλέξουμε με πιο mode θα υλοποιήσουμε την διαδικασία όλου του πρωτοκόλλου. Παρατηρώντας λοιπόν, στην διάθεσή μας έχουμε τρία mode το I2C1, I2C2, I2C3, όπου αυτά προσφέρουν τα οφέλη που προσδίδει το πρωτόκολλο και είναι η βάση για να διαμορφώσουμε τους καταχωρητές του I2C σύμφωνα με όσα θέλουμε να κάνουμε. Τα χαρακτηριστικά που προσφέρει το πρωτόκολλο είναι η επιλογή της διεύθυνσης αν επιθυμούμε να είναι 7 bit ή 10 bit, έχουμε τα mode για την ταχύτητα της επικοινωνίας fast-mode, standard mode, fast-mode Plus, επιπλέον το αυτόνομο ρολόι που έχει το I2C μέσα στον μικροελεγκτή που ορίζεται, τις επιλογές να ξυπνάει ο master/slave από ένα

stop mode και το SMBus/PMBus, όπου η τεχνολογία του βασίζεται κυρίως στην αποτελεσματική διαχείριση της μπαταρίας των συσκευών όπως για παράδειγμα laptops, smart phones, κάμερες και άλλα. Όπως βλέπετε και στην εικόνα, η οποία είναι στο datasheet του μικροελεγκτή, δεν μπορούν όλα τα modes να υποστηρίξουν τα οφέλη του πρωτοκόλλου γι αυτό και η επιλογή μου στην διαμόρφωση των καταχωρητών είναι το I2C1 που μπορεί να υποστηρίξει τα πάντα.

Η πρώτη λειτουργία που γίνεται στην αρχικοποίηση του πρωτοκόλλου είναι το λεγόμενο software reset. Με τον τρόπο αυτό οι γραμμές SDA και SCL απελευθερώνουν ότι δεδομένα υπάρχουν κολλημένα. Τόσο τα control bits αλλά και τα status bits έρχονται στις αρχικές τιμές τους. Όπως είναι λογικό η κίνηση αυτή επηρεάζει δύο βασικούς καταχωρητές που είναι οι I2C_CR2 και I2C_ISR, οι οποίοι διαχειρίζονται τα bits που αφορούν μια επικοινωνία όπως για παράδειγμα το ACK/NACK bit, start και stop condition , διάφορα flag για τον έλεγχο των εισερχόμενων και εξερχόμενων bit σε πίνακες που αποθηκεύονται τα μηνύματά μας όπως BUSY, TXE RXNE και πολλά άλλα. Σύμφωνα με τις οδηγίες του μικροελεγκτή αυτό γίνεται μηδενίζοντας το bit PE για 3 συνεχόμενους παλμούς του ρολογιού. Το PE είναι το πρώτο bit του καταχωρητή I2C_CR1 δηλαδή control register 1 και είναι υπεύθυνο για την ενεργοποίηση και απενεργοποίηση του περιφερειακού. Γιατί ωστόσο χρειάζεται να γίνει αυτό; Το πρωτόκολλο I2C παρακολουθεί τον δίαυλο επικοινωνίας (bus) κάθε φορά που χρονίζεται από το RCC (Reset and Clock Control register). Αν κατά την διάρκεια της ακολουθίας ενεργοποίησης του πρωτοκόλλου η κατάσταση του διαύλου (bus) στο I2C δεν είναι καλά καθορισμένη, όσο αναφορά με τις αρχικές τιμές τόσο των registers αλλά και των γραμμών SDA και SCL, τότε το περιφερειακό μπορεί να λάβει την ακολουθία ενεργοποίησης που εκτελούμε ως δραστηριότητα του διαύλου, δηλαδή νομίζει ότι μεταφέρονται δεδομένα. Έτσι λοιπόν ψάχνει συνεχώς στον δίαυλο για ένα stop condition. Το πρόβλημα εδώ είναι ότι αν ξεκινήσουμε την διαδικασία αποστολής με την ενέργεια του start condition τότε θα αποτύχει και αυτό γιατί την στιγμή που γίνεται το start condition εκείνο θεωρεί ότι ο δίαυλος είναι απασχολημένος.

I2C features ⁽¹⁾	I2C1	I2C2	I2C3
7-bit addressing mode	X	X	X
10-bit addressing mode	X	X	X
Standard-mode (up to 100 kbit/s)	X	X	X
Fast-mode (up to 400 kbit/s)	X	X	X
Fast-mode Plus with 20mA output drive I/Os (up to 1 Mbit/s) ⁽²⁾	X	X	X
Independent clock	X	-	X
Wakeup from Stop mode	X	-	X
SMBus/PMBus	X	-	X

1. X = supported.

2. Refer to the datasheet for the list of I/Os that support this feature.

Εικόνα 31 Χαρακτηριστικά πρωτοκόλλου I2C

Παρατηρώντας προσεκτικά το block diagram που είναι διαθέσιμο για τον μικροελεγκτή μπορούμε εύκολα να διαπιστώσουμε ότι το mode I2C1 είναι άμεσα συνδεδεμένο με το APB1 bus, το οποίο είναι υπεύθυνο για την παροχή του ρολογιού στο συγκεκριμένο mode. Έτσι λοιπόν ενεργοποιούμε το ρολόι στο I2C1 χρησιμοποιώντας τον clock control register (CCR). Πιο συγκεκριμένα βρίσκω τον καταχωρητή RCC_APB1ENR και συγκεκριμένα το 21 πρώτο bit του που ονομάζεται I2C1EN και το ενεργοποιώ γράφοντάς του την τιμή 1. Η επόμενη κίνηση είναι η ενεργοποίηση του automatic end bit που είναι διαθέσιμο στον καταχωρητή I2C_CR2 control register 2. Τι είναι το συγκεκριμένο bit που ενεργοποιήθηκε και ποιός ο ρόλος του; Το πρωτόκολλο I2C έχει ενσωματωμένο έναν μετρητή για τα bytes, όπου χρησιμοποιείται μόνο στην λειτουργία master, ωστόσο είναι διαθέσιμο και για slave μόνο αν το διαμορφώσουμε μέσω του κώδικα μας αλλάζοντας τις τιμές των bit κάποιων καταχωρητών. Τα δεδομένα που είναι για μεταφορά όταν θέλουμε να γράψουμε σε ένα περιφερειακό βρίσκονται στον πίνακα NBYTES που ελέγχεται από τον καταχωρητή I2C_CR2. Όταν ο μετρητής φτάσει στο τελευταίο byte που είναι για μεταφορά ενεργοποιεί το automatic end το οποίο είναι υπεύθυνο να στέλνει ένα stop condition αυτόματα μόλις τελειώσουν τα δεδομένα που μεταφέρονται προς το περιφερειακό που θεωρείται ως slave. Προχωρώντας με την αρχικοποίηση του I2C από τα πιο σημαντικά πράγματα που πρέπει να ορίσουμε είναι ο χρονισμός του πρωτοκόλλου. Η ST παρέχει ένα χρήσιμο εργαλείο που μας βοηθά να ορίσουμε με πιο εύκολο τρόπο τον χρονισμό του I2C από το να ασχοληθούμε σε βάθος με τους καταχωρητές του TIMINGR register και τις μαθηματικές τους εξισώσεις. Έτσι λοιπόν το εργαλείο αυτό ορίζοντας τις αρχικοποιήσεις που έχουμε θέσει στο πρωτόκολλο και ανάλογα με τα MHz του ρολογιού μας, μας αποδίδει μια αρκετά προσεγγιστική τιμή που μπορούμε να την

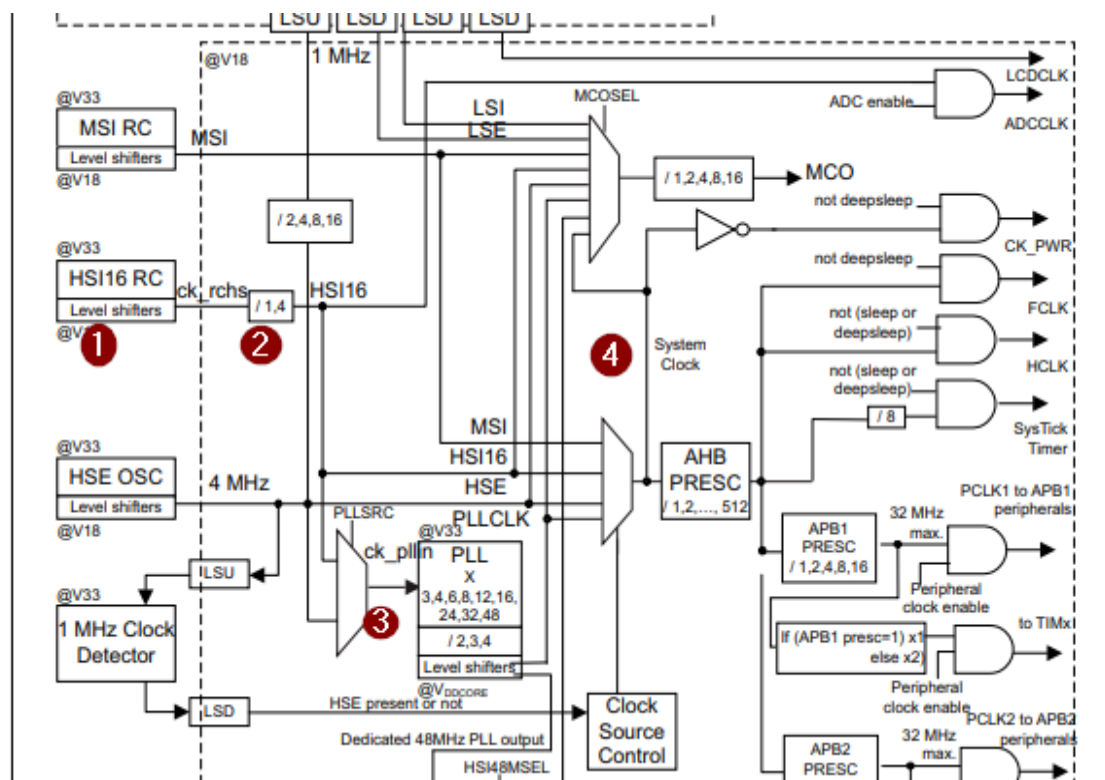
γράψουμε απλά στον TIMINGR register και να γίνουν οι απαραίτητες και πιο ενδελεχείς πράξεις. Οπότε για την τιμή του ρολογιού στα 32MHz το εργαλείο της ST, που ονομάζεται I2C timing config tool και είναι ένα excel αρχείο που διατίθεται στην ιστοσελίδα της ST, μας δίνει την προσεγγιστική τιμή 0x00805C89 που πρέπει να γραφτεί στον απαραίτητο register για τον χρονισμό του I2C. Συνεχίζοντας με την αρχικοποίηση το επόμενο βήμα είναι ο καθορισμός της διεύθυνσης του slave αλλά και της λειτουργίας που θα πράττει συνεχώς ο master δηλαδή ο μικροελεγκτής που είναι η εγγραφή δεδομένων προς το περιφερειακό μιας και μόνο αυτό χρειαζόμαστε για την λειτουργικότητα του συστήματος που έχουμε φτιάξει. Τέλος ενεργοποιείται το περιφερειακό και το σύστημα είναι έτοιμο από την πλευρά της αρχικοποίησης του.

4.5 Μείωση κατανάλωσης – Sleep Mode

Η μείωση κατανάλωσης του συστήματος χωρίζεται σε δύο φάσεις. Αρχικά καλείται η συνάρτηση που προετοιμάζει το σύστημα του μικροελεγκτή ώστε να εισέλθει στην κατάσταση χαμηλής κατανάλωσης και αυτή είναι η `<void prepare_low_power(void)>`. Μειώνονται οι παλμοί του ρολογιού από 32 MHz σε 2MHz και έπειτα οδηγείται ο μικροελεγκτής στην κατάσταση sleep mode, ενώ ταυτόχρονα απενεργοποιείται το led της οθόνης LCD μέσω ειδικής συνάρτησης της LCD. Για την μείωση των παλμών ρολογιού έγινε η αλλαγή στο ρολόι του συστήματος, όπου χρησιμοποιήθηκε το ρολόι MSI. Το σήμα του να παράγεται από έναν εσωτερικό RC ταλαντωτή. Οι συχνότητα του είναι εξ' ορισμού στα 2.097 MHz, οπότε με αυτόν τον τρόπο επιτυγχάνεται η άμεση πτώση του ρολογιού. Για την αλλαγή ωστόσο, σύμφωνα με τις οδηγίες της κατασκευαστικής εταιρίας απαιτείται μια διαδικασία όπου πρώτα ελέγχεται αν το ρολόι που έχει τεθεί ως ρολόι συστήματος είναι ενεργό και έπειτα ενεργοποιείται το MSI γράφοντας τα δύο πρώτα bit του καταχωρητή RCC_CFGR, ως "00", που ονομάζονται SW και είναι υπεύθυνα για τον έλεγχο του ρολογιού στο σύστημα μέσω προγραμματισμού. Η ασφαλής ενεργοποίηση της κατάστασης sleep mode, επιτυγχάνεται μέσω ενός καταχωρητή που ονομάζεται CSR και ανήκει στην οικογένεια καταχωρητών System Control Block (SCB), όπου το SCB παρέχει πληροφορίες υλοποίησης αλλά και την δυνατότητα ελέγχου του συστήματος. Έτσι λοιπόν ο SCR καταχωρητής είναι υπεύθυνος τόσο για την είσοδο όσο και για την έξοδο του μικροελεγκτή στην κατάσταση χαμηλής κατανάλωσης sleep mode.

4.6 Ρολόι συστήματος

Για το ρολόι συστήματος εκτελείται μια διαδικασία που έχει ως αφετηρία το εσωτερικό ρολόι HSI High Speed Internal clock. Η διαδικασία για την ενεργοποίηση του βασίστηκε στο εξής βοηθητικό διάγραμμα από το datasheet.



Εικόνα 32 Διάγραμμα ρολογιού (Clock tree) του μικροελεγκτή.

Όπως παρατηρείται από την παραπάνω εικόνα από το clock tree του μικροελεγκτή ενεργοποιείται με τον κατάλληλο register (RCC, Reset and Clock Control) το HSI16 οπότε σε αυτή την περίπτωση τίθενται τα 16MHz του ρολογιού χωρίς ακόμα να γίνει ρολόι συστήματος. Το HSI16 είναι εσωτερικό clock signal του μικροελεγκτή και μπορεί να χρησιμοποιηθεί ως ρολόι συστήματος ή ως είσοδος στο PLL. Στην συνέχεια επιλέγεται η εφαρμογή του διαιρέτη HSIDIVEN ώστε να πραγματοποιηθεί η μείωση των MHz με σκοπό στην είσοδό του στο PLL να γίνουν οι κατάλληλες πράξεις ώστε να φτάσει να δουλεύει το ρολόι του συστήματος στα 32MHz. Την στιγμή που το ρολόι εισέρχεται στο PLL έχει 4 MHz οπότε διαιρώντας με το 2 και πολλαπλασιάζοντας με το 16 ορίζεται το επιθυμητό αποτέλεσμα για το ρολόι του συστήματος που είναι 32MHz. Στο σημείο αυτό είναι σημαντικό να αναφερθεί πως αν δεν γίνει το HSI16 system clock από την αρχή, αλλά χρησιμοποιηθεί ως είσοδος στο PLL πρέπει το PLL να ενεργοποιηθεί αφού γίνουν οι απαραίτητες πράξεις για τον χρονισμό του ρολογιού. Το αντίστοιχο

παράδειγμα κώδικα είναι διαθέσιμο στο κεφάλαιο <Παράρτημα Α> στην ενότητα <Ενεργοποίηση ρολογιού PLL ως ρολόι συστήματος-RCC.

Κεφάλαιο 5. Συζήτηση – Συμπεράσματα

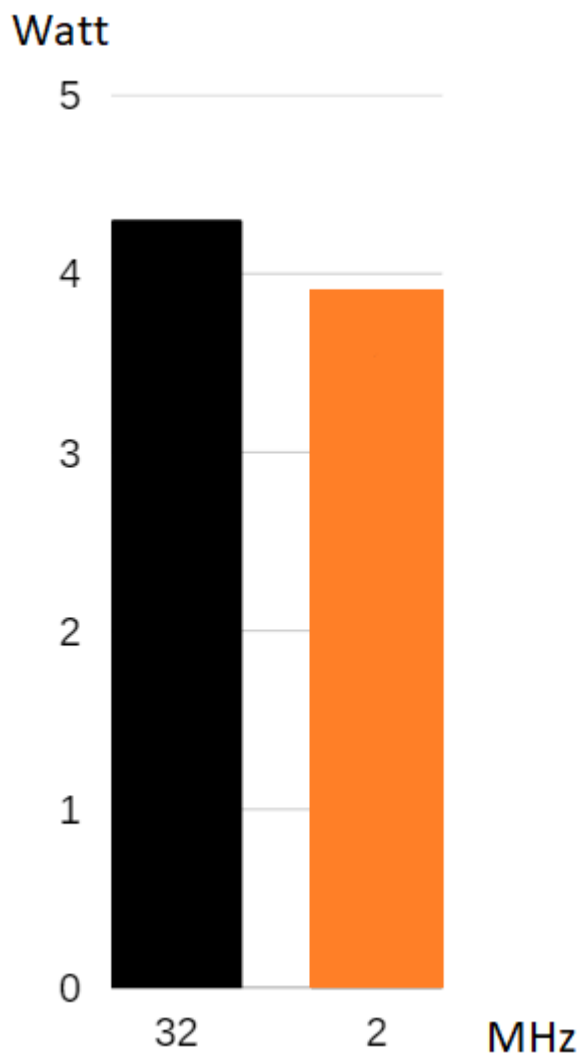
Στο παρόν κεφάλαιο γίνεται ανάλυση των πειραμάτων σχετικά με την μείωση της κατανάλωσης του συστήματος καθώς επίσης και η εξαγωγή συμπερασμάτων τόσο για την αποτελεσματικότητα των αποτελεσμάτων. Η μέτρηση της κατανάλωσης στο σύστημα πραγματοποιήθηκε με την βοήθεια ενός πολυμέτρου. Έχοντας, μια σταθερή πηγή τάσης ρεύματος στα 5 V, ξεκίνησε η διαδικασία μετρήσεων για την ισχύ που παράγει το σύστημα. Οι μετρήσεις της έντασης ρεύματος διαφέρουν ανάλογα με την κατάσταση στην οποία βρίσκεται ο μικροελεγκτής. Για παράδειγμα στην κανονική (normal) λειτουργία παράγει 0.86 μA ηλεκτρικής ισχύς για 32MHz συχνότητα ρολογιού. Αντίστοιχα στις φάσεις των δύο πειραμάτων, έχουμε 0.8 μA για το πρώτο και 0.55 μA για το δεύτερο. Όταν το σύστημα επικοινωνίας λειτουργεί σε κανονική (normal) κατάσταση, έχοντας δηλαδή 32 MHz συχνότητα ρολογιού η ισχύς του συστήματος είναι στα 4,3 P. Ωστόσο το αποτέλεσμα και των δύο πειραμάτων οδήγησε το σύστημα στο να εξοικονομεί ενέργεια όπου η κατανάλωση ισχύος την στιγμή εκείνη είναι 2.85 P. Στον παρακάτω πίνακα διακρίνονται οι τιμές των μετρικών για κάθε πείραμα.

	Ταχύτητα ρολογιού	Κατανάλωση
Κανονική λειτουργία	32 MHz	4.3 Watt
Πείραμα 1	2 MHz	3.9 Watt
Πείραμα 2	2 MHz	2.85 Watt

Πείραμα 1 Μείωση των παλμών ρολογιού

Ξεκινώντας, στο πρώτο πείραμα πραγματοποιήθηκε η μείωση της ταχύτητας στο ρολόι του συστήματος και από 32 MHz οδηγήθηκε στα 2MHz, ενεργοποιώντας το

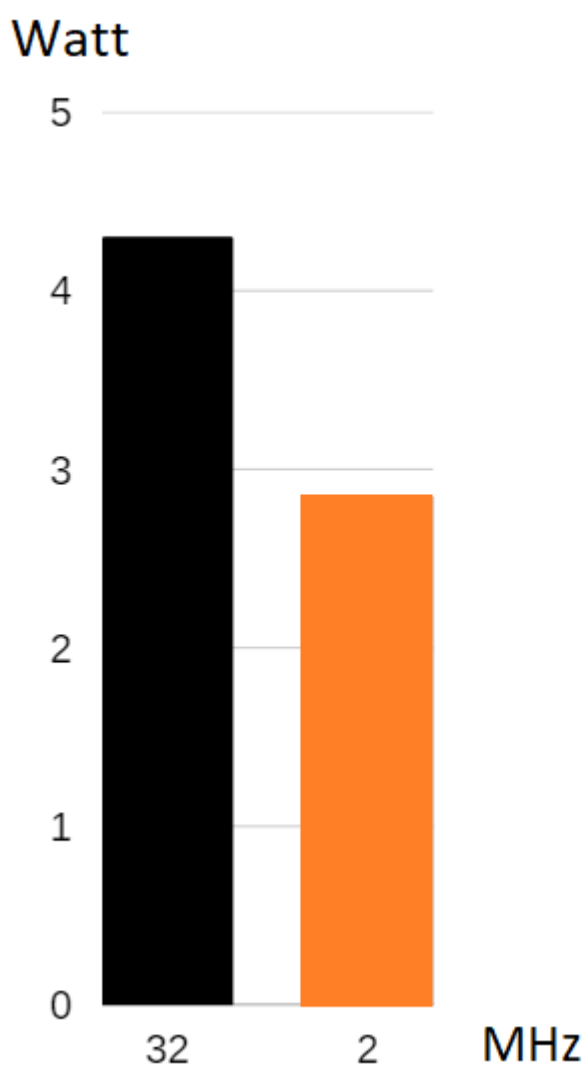
ρολίοι MSI. Η επιλογή αυτή οδήγησε το σύστημα στο να εξοικονομήσει ενέργεια ωστόσο το αποτέλεσμα δεν είναι αρκετά ικανοποιητικό καθώς η πλακέτα του μικροελεγκτή καταναλώνει ενέργεια από αρκετά πράγματα όπως το αυτόνομο κομμάτι που έχει ενδείξεις LED για την ενημέρωση του χρήστη αλλά και για τα περιφερειακά του όπως η οθόνη αλλά και ο ασύρματος πομποδέκτης που και εκείνα με την σειρά τους καταναλώνουν ενέργεια αφού αξιοσημείωτο αποτελεί το παράδειγμα του LED της οθόνης του περιφερειακού που παραμένει ανοικτό στο πρώτο πείραμα. Η συγκεκριμένη μεταβολή στην εξοικονόμηση ενέργειας διακρίνεται και στις παρακάτω εικόνες γραφημάτων.



Εικόνα 33 Μείωση απόδοσης ισχύος σύμφωνα με το πρώτο πείραμα

Πείραμα 2 Ενεργοποίηση low power κατάστασης και απενεργοποίηση περιφερειακών

Το επόμενο πείραμα είναι πιο σύνθετο από το πρώτο καθώς πραγματοποιείται η είσοδος του μικροελεγκτή στην κατάσταση χαμηλής κατανάλωσης low power και ταυτόχρονα απενεργοποιούνται τα περιφερειακά του όπως είναι το LED που αναβοσβήνει στο σύστημα για την ένδειξη της σωστής λειτουργίας, το LED της οθόνης LCD καθώς επίσης και η λειτουργία του ασύρματου πομποδέκτη για ένα συγκεκριμένο χρονικό όριο. Η ενέργεια σε συνδυασμό με το πρώτο πείραμα οδήγησε την καλύτερη εξοικονόμηση του συστήματος, που θα μπορούσε να επιτευχθεί με τον τρόπο που επιλέχθηκε να γίνει, αφού πραγματοποιήθηκε μείωση στην κατανάλωση στα 33 mA.

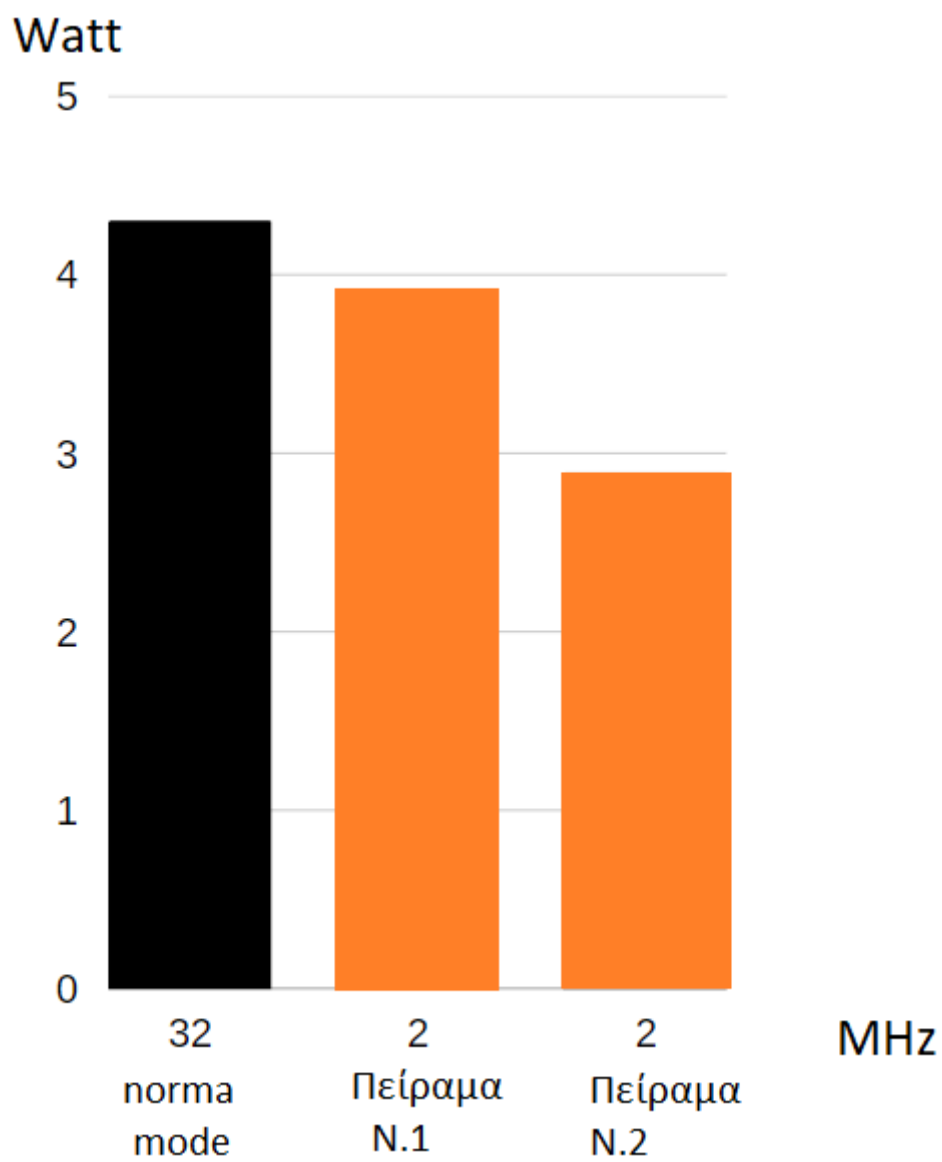


Εικόνα 34 Μείωση της απόδοσης ισχύος με βάση το δεύτερο πείραμα

Συμπέρασμα

Η πλακέτα του μικροελεγκτή όπως προαναφέρθηκε έχει αρκετά πράγματα που καταναλώνουν ενέργεια και τα οποία δεν γίνεται να ελεγχθούν ώστε να πραγματοποιηθεί οποιαδήποτε τροποποίηση που θα μας οδηγήσει στην μείωση της κατανάλωσης. Αρχικά υπάρχουν κάποιες φωτεινές ενδείξεις LED όπου ενημερώνουν την σωστή λειτουργία του κυκλώματος αλλά και την ροή ρεύματος στο σύστημα. Έπειτα αρκετή κατανάλωση προέρχεται και από το κομμάτι του προγραμματιστή της

πλακέτας όπου προγραμματίζει τον μικροελεγκτή μας κάθε φορά που πραγματοποιούμε μια νέα αλλαγή στον κώδικά μας και ταυτόχρονα μια νέα ενημέρωση προς τον μικροελεγκτή. Ωστόσο σημαντικό παράγοντα για τις μικρές αλλαγές στην κατανάλωση διαδραματίζει και ο ίδιος ο μικροελεγκτής καθώς είναι σχεδιασμένος ήδη να λειτουργεί σε εξαιρετικά χαμηλές επιδόσεις ενέργειας. Τέλος για μια πιο ειδική εικόνα στην μείωση της κατανάλωσης θα μπορούσε να αποτελέσει ένα πιο ειδικό εργαλείο για τις συγκεκριμένες μετρήσεις όπως για παράδειγμα ένα πολύμετρο με καλύτερη ακρίβεια ή γενικότερα εξειδικευμένα εργαλεία ώστε κάθε προγραμματιστική ενέργεια και πράξη να αποδώσει καρπούς και να φανεί στην εξοικονόμηση, καθώς με όχι και τόσο εξειδικευμένα εργαλεία οποιαδήποτε διαφορά, αφού θα είναι αρκετά μικρή δεν θα γίνεται αντιληπτή. Παρακάτω όπως θα δείτε παρουσιάζεται μια γενική εικόνα για την μείωση της κατανάλωσης που επιτεύχθηκε από την πορεία των πειραμάτων.



Εικόνα 35 Συνολική εικόνα απόδοσης πειραμάτων

Παράρτημα Α Παραδείγματα κώδικα

Το κεφάλαιο <Παράρτημα Α> παρουσιάζει παραδείγματα κώδικα που αναφέρθηκαν στο αντίστοιχο κεφάλαιο <Υλοποίηση>. Η δομή του κώδικα είναι βασισμένη στην βιβλιοθήκη CMSIS με κύριο αρχείο κεφαλίδας να είναι το stm32l0xx.h.

A.1 Ενεργοποίηση ρολογιού PLL ως ρολόι συστήματος-RCC.

```
/* (1) Ενεργοποίηση του εσωτερικού ταλαντωτή HSI16 */  
/* (2) Αναμονή μέχρι να ολοκληρωθεί η ενεργοποίησή του */  
/* (3) Ενεργοποίηση του διαιρέτη ώστε να γίνει η πράξη 16/4 και να μειωθεί  
το ρολόι του συστήματος στα 4MHz, όπως φαίνεται και στην εικόνα , βήμα 2 */  
/* (4) Αναμονή μέχρι να γίνει η πράξη που σημειώθηκε. Η ολοκλήρωσή της  
σηματοδοτείται από την σημαία “divider flag” που είναι το bit  
“RCC_CR_HSIDIVF” του καταχωρητή “RCC_CR”. */  
/* (5) Ολική εκκαθάριση, αποφυγή trimming. */  
  
    Πράξεις στο PLL  
/* (6) PLLDIV 4MHz/2 = 2 MHz. */  
/* (7) PLLMUL 2MHz*16 = 32MHz. */  
/* (8) Ενεργοποίηση του ρολογιού PLL. */  
/* (9) Αναμονή μέχρι να ολοκληρωθεί η διαδικασία. */  
/* (10) Ορισμός PLL ως ρολόι συστήματος. */  
/* (11) Αναμονή για την ολοκλήρωση της παραπάνω ενέργειας. */  
/* (12) Ανανέωση της μεταβλητής που έχει τεθεί στην διαδικασία του  
debugging, για τον έλεγχο της ορθής λειτουργίας */
```

```

    / * HSI16 & PLL => 32MHz clock */
RCC->CR |= RCC_CR_HSION; /*(1)*/
while(!(RCC->CR & RCC_CR_HSIRDY)); /*(2)*/
RCC->CR |= RCC_CR_HSIDIVEN; /*(3)*/
while(!(RCC->CR & RCC_CR_HSIDIVF)); /*(4)*/
RCC->ICSCR = 0x0000B000; /*(5)*/
RCC->CFGR |= RCC_CFGR_PLLDIV_0; /*(6)*/
RCC->CFGR |= (RCC_CFGR_PLLMUL_2 | RCC_CFGR_PLLMUL_0); /*(7)*/
RCC->CR |= RCC_CR_PLLON; / * (8) * /
while((RCC->CR & RCC_CR_PLLRDY) != RCC_CR_PLLRDY); /*(9)*/
RCC->CFGR |= RCC_CFGR_SW; /*(10)*/
while((RCC->CFGR & RCC_CFGR_SWS) != RCC_CFGR_SWS); /*(11)*/
SystemCoreClockUpdate(); /*(12)*/

```

A.2 Low power – Sleep mode

```

/ *(1) Απενεργοποίηση κάθε διακοπής από τον μετρητή SysTick. */
/ *(2) Απενεργοποίηση της λειτουργίας sleep on exit */
/ *(3) Απενεργοποίηση της κατάστασης deep sleep. */
/ *(4) Είσοδος στην κατάσταση χαμηλής κατανάλωσης και αναμονή για σήμα
διακοπής, ώστε να επανέλθει ο μικροελεγκτής σε κανονική λειτουργία. */
/ *(5) Επαναφορά ρυθμίσεων μετρητή SysTick μετά την αφύπνιση. */
CLEAR_BIT(SysTick->CTRL,SysTick_CTRL_TICKINT_Msk);/*(1)*/
CLEAR_BIT(SCB->SCR,SCB_SCR_SLEEPONEXIT_Msk);/*(2)*/
CLEAR_BIT(SCB->SCR,SCB_SCR_SLEEPDEEP_Msk);/*(3)*/
__WFI(); / * (4) * /
SysTick->CTRL |= SysTick_CTRL_TICKINT_Msk; /*(5)*/

```

A.2.1 Μείωση παλμών ρολογιού – Ενεργοποίηση MSI clock.

```

/ *(1) Έλεγχος αν το ρολόι PLL δρα ως ρολόι συστήματος. */
/ *(2) Ορισμός του ρολογιού MSI ως ρολόι συστήματος. */
/ *(3) Αναμονή για την ολοκλήρωση της αλλαγής. */

```



```

/ *(4) Ανανέωση της μεταβλητής για τον έλεγχο του ρολογιού, όσον αναφορά την
μείωση στα MHz. */
if ((RCC->CFGR & RCC_CFGR_SWS) == RCC_CFGR_SWS_PLL){ /*(1)*/
    WRITE_REG(RCC->CFGR, RCC_CFGR_SW_MSI); /*(2)*/
    while (READ_BIT(RCC->CFGR, RCC_CFGR_SWS) != RCC_CFGR_SWS_MSI); /*(3)*/
}
SystemCoreClockUpdate(); /*(4)*/

```

A.2.2 Απενεργοποίηση LED οθόνης.

```

/ *(1) Αναζήτηση της διεύθυνσης LED από το διάγραμμα διευθύνσεων του
PCF8574A. */
/ *(2) Απενεργοποίηση του bit οθόνης, εκτελώντας την ειδική συνάρτηση
lcd_write ώστε να στείλει την εντολή στην αντίστοιχη διεύθυνση. */
#define LED 0x08 /*(4)*/
CLEAR_BIT(PINS, LED); /*(2)*/
i2c_write(); /*(2)*/

```

A.3 Αρχικοποίηση SPI

```

/ *(1) Ενεργοποίηση του ρολογιού SPI μέσω του διαύλου APB2. * /
/ *(2) Έλεγχος baud rate στα 8MHz*/
/ *(3) Ενεργοποίηση της επιλογής master mode. */
/ *(4) Software slave management enabled.*/
/ *(5) Ενεργοποίηση της επιλογής σήματος διακοπής σε περίπτωση εισερχόμενου
πακέτου. */
/ *(6) Απενεργοποίηση CSN ακροδέκτη. */
/ *(7) Ενεργοποίηση διακοπών πρωτοκόλλου SPI. */
/ *(8) Ενεργοποίηση πρωτοκόλλου SPI*/
SET_BIT(RCC->APB2ENR, RCC_APB2ENR_SPI1EN); /*(1)*/
SET_BIT(SPI1->CR1, SPI_CR1_BR_0); /*(2)*/
SET_BIT(SPI1->CR1, SPI_CR1_MSTR); /*(3)*/

```

```

SET_BIT(SPI1->CR1, (SPI_CR1_SSM | SPI_CR1_SSI)); /*(4)*/
SET_BIT(SPI1->CR2, SPI_CR2_RXNEIE); /*(5)*/
SPI_RX_COUNTER = 0;
SPI_TX_SIZE = 0;
CSN_PIN_DIS(); /*(6)*/
NVIC_EnableIRQ(SPI1_IRQn); /*(7)*/
SET_BIT(SPI1->CR1, SPI_CR1_SPE); /*(8)*/

```

A.3.1 SPI λήψη και αποστολή δεδομένων με interrupt

```

/ *(1) Σήμα διακοπής αν ο πίνακας ληφθέντων δεδομένων δεν είναι γεμάτος. * /
/ *(2) Εγγραφή δεδομένων στον πίνακα RX.* /
/ *(3) Σήμα διακοπής όταν ο πίνακας εξερχόμενων μηνυμάτων είναι γεμάτος.
*/
/ *(4) Απενεργοποίηση NSS ακροδέκτη. * /
/ *(5) Απενεργοποίηση TX inderrupt.* /

```

```

if(READ_BIT(SPI1->SR, SPI_SR_RXNE)){ /*(1)*/
    SPI_RX_BUFFER[SPI_RX_COUNTER] = SPI1->DR; /*(2)*/
    SPI_RX_COUNTER ++;
}
if(READ_BIT(SPI1->SR, SPI_SR_TXE)){ /*(3)*/
    if(!SPI_TX_SIZE){
        CSN_PIN_DIS(); /*(4)*/
        CLEAR_BIT(SPI1->CR2, SPI_CR2_TXEIE); /*(5)*/
    }else{
        WRITE_REG(SPI1->DR, SPI_TX_BUFFER[SPI_TX_COUNTER]);
        SPI_TX_SIZE --;
        SPI_TX_COUNTER ++;
    }
}
}

```

A.3.2 Εξασφάλιση κατάστασης RX

```
/ *(1) Barrier σημαία(flag) που χρησιμοποιείται όταν γίνεται αποστολή και
λήψη δεδομένων, οπότε αν δεν γίνεται αποστολή/λήψη δεδομένων ξεκινάει η
διαδικασία καθοδήγησης του περιφερειακού στην κατάσταση RX. * /

/ *(2) Απενεργοποίηση CE ακροδέκτη ώστε να βρεθεί το περιφερειακό στην
κατάσταση STANDBY-I και να καθοριστούν οι απαραίτητες ρυθμίσεις. * /

/ *(3) Set RX payload payload.* /

/ *(4) Ενεργοποίηση κάθε είδους σήματος διακοπής στα εξής: CRC, POWE UP
PRIM_RX PRX_MODE. */

/ *(5) Εφαρμογή καθυστέρησης για την ενεργοποίηση των προδιαγραφών 1,5ms. *
/

/ *(6) Ενεργοποίηση του ακροδέκτη CE, ώστε να οδηγηθεί το περιφερειακό στην
κατάσταση RX με σκοπό να λάβει πακέτο μηνυμάτων.* /

/ *(7) Εφαρμογή barrier ώστε να μην μπορεί να γίνει ταυτόχρονη αποστολή και
λήψη δεδομένων από το ίδιο σύστημα.* /

//RX size must be equal to TX size
uint8_t nrf_receive(uint8_t size){
    if(!flag0.fl.f4){ /*(1)*/
        CE_DEACTIVATE();/*(2)*/
        nrf_write_reg(RX_PW_P0, size);/*(3)*/
        nrf_write_reg(CONFIG, 0x0B); /*(4)*/
        delay_ms(3); /*(5)*/
        CE_ACTIVATE();/*(6)*/
        flag0.fl.f4 = 1;/*(7)*/
        return 1;
    }
    return 0;
}
```

A.3.3 Διάβασμα ωφέλιμου φορτίου

```
/ *(1) Διάβασμα και αποθήκευση μήκους από το πακέτο ωφέλιμου φορτίου που θα
διαβαστεί, μέσω της SPI Command R_RX_PL_WID. * /
```

```

/ *(2) Στην πρώτη θέση του πίνακα φορτώνεται η εντολή μέσω της SPI Command
ώστε να ενημερωθεί το περιφερειακό για την ενέργεια που θα εκτελέσει. * /
/ *(3) Αποθήκευση στον πίνακα τόσα dummy bytes όσα και του μηνύματος που
πρέπει να διαβάσουμε.* /
/ *(4) Αποστολή του πίνακα, μέσω της συνάρτησης spi. */
/ *(5) Διάβασμα του ωφέλιμου φορτίου. * /
uint8_t nrf_read_payload(void){
    uint8_t payload_width = nrf_read_reg(R_RX_PL_WID);/*(1)*/
    uint8_t i;
    if(payload_width > 0){
        payload_data[0] = R_RX_PAYLOAD;/*(2)*/
        for(i=1;i<33;i++){
            payload_data[i] = NOP;/*(3)*/
        }
        spi_transmit_wait(payload_data, payload_width+1);/*(4)*/
        for(i=0;i<payload_width;i++){
            payload_data[i] = SPI_RX_BUFFER[i+1];/*(5)*/
        }
        return payload_width;
    }
    return 0;
}

```

A.4 Αρχικοποίηση I2C

```

/ *(1) Επαναφορά ρυθμίσεων (reset) στο I2C. * /
/ *(2) Ενεργοποίηση ρολογιού I2C μέσω APB1 διαύλου*/
/ *(3) Ενεργοποίηση Automatic end, ώστε ο master να στέλνει αυτόματο
κατάσταση τερματισμού μόλις γίνει η αποστολή ολόκληρου του ωφέλιμου
φορτίου. */
/ *(4) Καθορισμός διεύθυνσης στο περιφερειακό που βρίσκεται στην οθόνη
PCF8574. */
/ *(5) Ενεργοποίηση χρονισμού, σύμφωνα με τις ενδείξεις των κατασκευαστών,
για 32MHz*/

```

```

/ *(6) Ενεργοποίηση της λειτουργίας I2C. */
SET_BIT(RCC->APB1RSTR, RCC_APB1RSTR_I2C1RST);
CLEAR_BIT(RCC->APB1RSTR, RCC_APB1RSTR_I2C1RST);

SET_BIT(RCC->APB1ENR, RCC_APB1ENR_I2C1EN); /*(2)*/
SET_BIT(I2C1->CR2, I2C_CR2_AUTOEND); /*(3)*/
MODIFY_REG(I2C1->CR2, I2C_CR2_SADD, (PCF8574_ADDRESS <<
I2C_CR2_SADD_Pos)); /*(4)*/
WRITE_REG(I2C1->TIMINGR, 0x00805C89); /*(5)*/
SET_BIT(I2C1->CR1, I2C_CR1_PE); /*(6)*/

```

A.5 Αντιμετώπιση Switch debouncing

```

/* Πάτημα κουμπιού */
if(!flag0.f1.f6){
    if(!KEY2_READ()){
        if(key2_counter > 10)
            flag0.f1.f6 = 1; // R2 is HIGH
        key2_counter = 0;
    }else{
        if(key2_counter < 10) //10 ms sampling
            key2_counter ++;
    }
}

```

A.6 Αρχικοποίηση Enhanced Shockburst

```

/ *(1) Απενεργοποίηση CE ακροδέκτη ώστε να μην γίνει είσοδος σε κατάσταση
λήψης ή αποστολής. * /
/ *(2) Μέσω ειδικής εντολής FLUSH_TX (SPI Command) αδειάζεται η ουρά
αποστολής δεδομένων.*/
/ *(3) Μέσω ειδικής εντολής FLUSH_RX (SPI Command) αδειάζεται η ουρά λήψης
δεδομένων. */
/ *(4) Reset σε κάθε είδους σήματα διακοπών του πομποδέκτη.*/
/ *(5) Ενεργοποίηση διακοπών, CRC και κατάστασης POWER DOWN */

```

```

/ *(6) Ενεργοποίηση αυτόματου μηνύματος επιβεβαίωσης στο δίαυλο
επικοινωνίας 0. */

/ *(7) Ενεργοποίηση διαύλου 0 για την απομακρυσμένη επικοινωνία.*/

/ *(8) Ορισμός 3 byte για τις αντίστοιχες διευθύνσεις αποστολέα παραλήπτη.
*/

/ *(9) Ορισμός περιόδου, αλλά και αριθμών επαναπροσπαθειών αποστολής
μηνύματος, Retransmit every 2000us = 2ms, up to 15 retrasmits. */

/ *(10) Ορισμός συχνότητας στα 2.4GHz όπως αναφέρει και η κατασκευαστική
εταιρία.*/

/ *(11) Ορισμός 2Mbps data rate, 0dBm (max) transmit power.*/

/ *(12) Προσδιορισμός της διεύθυνσης παραλήπτη . */

/ *(13) Προσδιορισμός της διεύθυνσης αποστολέα . */

/ *(14) Ενεργοποίηση του δυναμικού μήκους ωφέλιμου φορτίου (dynamic payload
length, καθώς επίσης και την αυτόματη επιβεβαίωση πακέτων μηνυμάτων ACK,
στον δίαυλο 0. */

/ *(15) Ενεργοποίηση δυναμικού μήκους ωφέλιμου φορτίου στο δίαυλο
επικοινωνίας 0. */

CE_DEACTIVATE();/*(1)*/

nrf_send_command(FLUSH_TX);/*(2)*/

nrf_send_command(FLUSH_RX);/*(3)*/

nrf_write_reg(STATUS, 0x70);/*(4)*/

nrf_write_reg(CONFIG, 0x08);/*(5)*/

nrf_write_reg(EN_AA, 0x01);/*(6)*/

nrf_write_reg(EN_RXADDR, 0x01);/*(7)*/

nrf_write_reg(SETUP_AW, 0x01);/*(8)*/

nrf_write_reg(SETUP_RETR, 0x7F);/*(9)*/

nrf_write_reg(RF_CH, 0x00);/*(10)*/

nrf_write_reg(RF_SETUP, 0x0E);/*(11)*/

spi_transmit_wait(nrf_rx_pipe0_address, 4);/*(12)*/

spi_transmit_wait(nrf_tx_address, 4);/*(13)*/

nrf_write_reg(FEATURE, 0x06);/*(14)*/

nrf_write_reg(DYNPD,0x01);/*(15)*/

```

A.7 Προγραμματισμός εξωτερικού σήματος διακοπής

```
/ *(1) Ενεργοποίηση ρυθμίσεων συστήματος (system configuration controller).  
* /  
  
/ *(2) Εφαρμογή εξωτερικού σήματος διακοπής από τον ακροδέκτη στην γραμμή  
8.*/  
  
/ *(3) Ενεργοποίηση του σήματος διακοπής στο falling edge του παλμού  
ρολογιού. */  
  
/ *(4) Διαμόρφωση του ακροδέκτη PA8 ως πηγή παραγωγής διακοπών.*/  
  
/ *(5) Ενεργοποίηση του αντίστοιχου τμήματος διαχείρισης διακοπών interrupt  
handler που αφορά του αριθμούς pin από 4-15. */  
  
  
SET_BIT(RCC->APB2ENR, RCC_APB2ENR_SYSCFGEN);/*(1)*/  
SET_BIT(EXTI->IMR, EXTI_IMR_IM9);/*(2)*/  
SET_BIT(EXTI->FTSR, EXTI_FTSR_FT9);/*(3)*/  
MODIFY_REG(SYSCFG->EXTICR[2],SYSCFG_EXTICR3_EXTI9,  
SYSCFG_EXTICR3_EXTI9_PB);/*(4)*/  
NVIC_EnableIRQ(EXTI4_15_IRQn);/*(5)*/
```

Βιβλιογραφία

- [1] David Seal. ARM Architecture Reference Manual. Addison-Wesley, 2nd edition, 2000.
- [2] Markus Levy. The history of the ARM architecture: From inception to IPO. ARM IQ, 4(1), 2005.

The Definitive Guide to ARM® Cortex®-M0 and Cortex-M0+ Processors
Joseph Yiu.
- [3] Hitex, The Insider's Guide To The STM32 ARM Based Microcontroller | An Engineer's Introduction To The STM32 Series, Version 1.8, February 2009.
- [4] Texas Instruments, Understanding the I2C Bus, Application Report SLVA704, June 2015
- [5] Γ. Αλεξίου, Ν. Πετρέλης Μικροεπεξεργαστές και σχεδιασμός μικροϋπολογιστικών συστημάτων, εκδόσεις κλειδάριθμος, 2012
- [6] Nrf24l01+, Single Chip 2.4GHz Transceiver, Product Specification v1.0, 2008