



Α.Τ.Ε.Ι. ΚΡΗΤΗΣ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ  
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ:

**“ΓΕΦΥΡΩΝΟΝΤΑΣ ΣΕΙΡΙΑΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ**

**RS232 ΜΕΣΩ ΤΕΧΝΟΛΟΓΙΩΝ ΑΣΥΡΜΑΤΩΝ**

**ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ”**

SUBJECT:

**“BRIDGING RS232 SERIAL DATA**

**COMMUNICATIONS OVER WIRELESS SENSOR**

**NETWORK TECHNOLOGIES”**

ΑΝΑΠΤΥΞΗ: Νικόλαος Πλευριτάκης

ΕΠΙΒΛΕΠΩΝ: Δρ. Ιωάννης Μπαρμπουνάκης

## ΕΥΧΑΡΙΣΤΙΕΣ

Η εργασία αυτή αφιερώνεται στους γονείς μου μαζί με ένα μεγάλο ευχαριστώ για την αγάπη και την πολύτιμη συμπαράστασή τους κατά τη διάρκεια των σπουδών μου.

Καταρχήν θα ήθελα να ευχαριστήσω τον καθηγητή Δρ. Ιωάννη Μπαρμπουνάκη για την καθοδήγηση, υποστήριξη και άψογη συνεργασία που είχα μαζί του ώστε να εκπληρωθεί αυτή η πτυχιακή εργασία σε περιορισμένο χρόνο όπως και για τις γνώσεις που μας μετέδωσε, σ'εμένα και τους συμφοιτητές μου καθ' όλη διάρκεια των σπουδών μου.

Τέλος ευχαριστώ τους φίλους, συμφοιτητές και συνάδελφους ηλεκτρονικούς Τηλέμαχο Παπαθεοδώρου, Κώστα Φραγκιαδάκη, Νίκο Δαμουρά και τον νεότερο Σάββα Τζαβάρα που είχα την τύχη να γνωρίσω και να μοιραστώ την αδυναμία μου, τον κόσμο της Ηλεκτρονικής.

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ – ABSTRACT.....	1
ΕΙΣΑΓΩΓΗ.....	2
ΜΕΤΑΦΟΡΑ ΔΕΔΟΜΕΝΩΝ.....	5
Σειριακή Επικοινωνία.....	6
Ασύγχρονη Σειριακή Επικοινωνία.....	7
Το Πρότυπο RS-232C.....	8
Τύποι Ασύγχρονων Σειριακών Συσκευών.....	10
Οι Ακροδέκτες της Σειριακής Θύρας και οι Λειτουργίες τους.....	11
Το CC2430 SoC.....	13
Μπλοκ Διάγραμμα & Χαρακτηριστικά.....	14
Ταλαντωτές.....	16
CPU.....	17
RAM (Data, iData, Xdata).....	17
ROM (Flash).....	18
Memory Crossbar - Arbitrator.....	19
DMA.....	19
Power Management Control.....	20
WDT.....	22
USARTs.....	22
RADIO.....	26
SIMPLICITI.....	30
Κατηγορίες κόμβων δικτύου.....	31
Τοπολογίες Δικτύου.....	34
Τα επίπεδα της στοίβας.....	35
ΠΕΙΡΑΜΑΤΙΚΗ ΕΦΑΡΜΟΓΗ.....	37

ΣΥΜΠΕΡΑΣΜΑΤΑ.....	41
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	42
ΠΑΡΑΡΤΗΜΑ .....	43
ΠΑΡΑΡΤΗΜΑ 2 .....	61

## ΠΕΡΙΛΗΨΗ – ABSTRACT

### ΓΕΦΥΡΩΝΟΝΤΑΣ ΣΕΙΡΙΑΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ RS232 ΜΕΣΩ ΤΕΧΝΟΛΟΓΙΩΝ ΑΣΥΡΜΑΤΩΝ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ

Η παρούσα εργασία επικεντρώνεται στην εφαρμογή τεχνολογιών ασύρματων δικτύων αισθητήρων για την μεταφορά σειριακών δεδομένων επεκτείνοντας την εμβέλεια της ενσύρματης σύνδεσης και διασφαλίζοντας παράλληλα την αξιοπιστία της. Πιο συγκεκριμένα, η εφαρμογή βασίζεται στην στοίβα SimpliCI TI και χρησιμοποιεί το SoC CC2430 με μικροελεγκτή αρχιτεκτονικής 8051 και πομποδέκτη 2.4GHz, που είναι συμβατός με το πρωτόκολλο IEEE 802.15.4.

Η διασύνδεση των απομακρυσμένων κόμβων για την μεταφορά των σειριακών δεδομένων είναι τύπου RS232 δεδομένου ότι εξακολουθεί να υιοθετείται από το μεγαλύτερο ποσοστό τηλεπικοινωνιακών διατάξεων, παρά την ευρεία αποδοχή της τεχνολογίας USB.

Επιπλέον, η εξάπλωση των τεχνολογιών ασύρματων δικτύων αισθητήρων και ιδιαίτερα το χαμηλό κόστος κατασκευής τους, προκάλεσε το ενδιαφέρον μας για αυτή την πειραματική εφαρμογή.

### BRIDGING RS232 SERIAL DATA COMMUNICATIONS OVER WIRELESS SENSOR NETWORK TECHNOLOGIES

There are many electronic devices still using the mature RS232 asynchronous serial communication whereas newer serial communication technologies are globally widespread such as USB v2.0, v3.0, Firewire etc. On the other hand, the hardware for Wireless Sensor Network Technologies is getting cheaper thus a new challenge application has been generated. Is it possible to “Bridge” RS232 communications over WSN links to extend its limited range over the legacy wired cable solutions? This also could add flexibility and mobility to such mature wire communications.

## ΕΙΣΑΓΩΓΗ

Στην τελευταία δεκαετία η ανάπτυξη της τεχνολογίας των αισθητήρων οδήγησε στην ανάπτυξη της κατασκευής χαμηλού κόστους, χαμηλής κατανάλωσης ενέργειας, πολυλειτουργικών και μικροσκοπικών αισθητήρων όπου οδήγησε σε πολλές νέες σύγχρονες τεχνολογικές εφαρμογές. Η συλλογή δεδομένων από το περιβάλλον έδωσε μεγάλη ώθηση στον αυτοματισμό και στην επιστημονική παρατήρηση. Παράλληλα με την ραγδαία ανάπτυξη της τεχνολογίας των ασύρματων επικοινωνιών και των μικροσυστημάτων οδήγησε σε μια καινοτομία με συγχώνευση της τεχνολογίας των αισθητήρων, των υπολογιστικών μικροσυστημάτων και των ασύρματων επικοινωνιών σε αυτό που καθιερωθεί ως Ασύρματα Δίκτυα Αισθητήρων. Η ανάπτυξη της τεχνολογίας των ασύρματων επικοινωνιών έκανε πιο αποτελεσματική και πιο οικονομική την χρήση των ασύρματων μέσων αντί των ενσύρματων μέσων στα δίκτυα αισθητήρων.

Γενικά, τα ασύρματα δίκτυα αισθητήρων είναι μια νέα τεχνολογία που έχει στόχο την παρακολούθηση και τον έλεγχο του φυσικού κόσμου. Τελευταία, αποτελούν την σπουδαιότερη εξέλιξη των ασύρματων επικοινωνιών. Ένα ασύρματο δίκτυο αισθητήρων είναι ένα δίκτυο υπολογιστικών συστημάτων, που αποτελείται από αυτόνομες συσκευές κατανεμημένες στο χώρο και οι οποίες χρησιμοποιούν αισθητήρες με σκοπό τη συλλογική μέτρηση και απεικόνιση φυσικών ή περιβαλλοντικών μεγεθών, όπως η θερμοκρασία, ο ήχος, η δόνηση, η πίεση, η κίνηση ή τα σωματίδια ρύπων, σε διάφορες τοποθεσίες. Οι εφαρμογές των δικτύων αισθητήρων μπορούν να ομαδοποιηθούν σε στρατιωτικές, υγείας, περιβάλλοντος, οικιακές και εμπορικές. Είναι δυνατόν να επεκτείνουμε την ομαδοποίηση σε περισσότερες κατηγορίες όπως εξερεύνηση του διαστήματος, χημική επεξεργασία, αντιμετώπιση καταστροφών κ.ά.

Αναλυτικότερα, τα ασύρματα δίκτυα αισθητήρων αποτελούνται από ένα ή περισσότερα sink (ή base station) και από μερικές δεκάδες ή εκατοντάδες κόμβους αισθητήρες (sensor nodes), οι οποίοι είναι διασκορπισμένοι σε ένα χώρο. Οι κόμβοι αυτοί συλλέγουν πληροφορίες από το περιβάλλον και ανάλογα με την εφαρμογή, είτε επεξεργάζονται τις πληροφορίες και τις στέλνουν, είτε τις στέλνουν χωρίς καμία επεξεργασία. Οι κόμβοι αυτοί, συνήθως πρέπει να αισθάνονται τη θερμοκρασία, το φως, τη δόνηση, τον ήχο, την ακτινοβολία κ.ά. Οι πληροφορίες αυτές «ταξιδεύουν» μέσα στο δίκτυο, έχοντας σαν τελικό προορισμό τους κόμβους sink. Ανάλογα με την εφαρμογή, τα sink ενδέχεται να αποστείλουν κάποια υποερωτήματα (queries) προς τους κόμβους, με σκοπό να μαζέψουν χρήσιμες πληροφορίες.

Το σημαντικό πλεονέκτημα, το οποίου παρέχει η εγκατάσταση ενός δικτύου ασύρματων αισθητήρων είναι ότι δεν απαιτείται η εκ των προτέρων γνώση της τοπολογίας του. Η δυνατότητα αυτή επιτρέπει την ταχεία ανάπτυξη δικτύων αυτού του τύπου σε δύσβατες ή ακατάλληλες για τον άνθρωπο περιοχές. Οι κόμβοι είναι χαμηλού κόστους και χαμηλής κατανάλωσης και έχουν τη δυνατότητα να επικοινωνούν σε μικρές αποστάσεις, να εκτελούν περιορισμένη τοπική επεξεργασία δεδομένων και να «αισθάνονται» διαφορών ειδών ερεθίσματα στην περιοχή εφαρμογής τους.

Ένα δίκτυο αισθητήρων χαρακτηρίζεται από το χρόνο ζωής του, την επεκτασιμότητα του, την κάλυψη που παρέχει, το κόστος παραγωγής του, την ευκολία ανάπτυξης, την ανίχνευση και διόρθωση σφαλμάτων, το χρόνο απόκρισης, τον τρόπο συγχρονισμού, αλλά και την ασφάλεια που μπορεί να παρέχει. Ο αναμενόμενος χρόνος ζωής του δικτύου είναι από τα σημαντικότερα χαρακτηριστικά και κυριότερος περιοριστικός παράγοντας στην διάρκεια ζωής του είναι η χωρητικότητα του συσσωρευτή ενέργειας του συστήματος.

Αμέσως επόμενοι σημαντικοί παράγοντες μετά τον χρόνο ζωής, είναι η κάλυψη και η επεκτασιμότητα. Είναι σημαντικό στον διαχειριστή να αναπτύξει δίκτυα τα οποία να καλύπτουν μια ευρεία περιοχή παρατήρησης. Η τεχνολογία ασύρματης δικτύωσης είναι εφικτή κάτω από κατάλληλο σχεδιασμό, η επέκταση της κάλυψης πολύ πιο μακριά από την ακτίνα εκπομπής του κόμβου. Μιας και τα δίκτυα αισθητήρων αποτελούνται από πολλούς κόμβους, επιδιώκεται το κόστος ενός κόμβου να είναι χαμηλό ώστε το συνολικό κόστος του δικτύου να είναι μια λογική τιμή.

Τα ασύρματα δίκτυα αισθητήρων παρουσιάζουν ένα ιδιαίτερο σύνολο από περιορισμούς όπως χαμηλή υπολογιστική ισχύς και μνήμη προγράμματος & δεδομένων μικρής χωρητικότητας. Ο πιο σημαντικός από αυτούς είναι η ενέργεια. Αυτά τα δίκτυα αποτελούνται από συσκευές που πρέπει να είναι ενεργές αρκετή ώρα με μικρές μπαταρίες. Κατανάλωση ενέργειας έχουμε:

- Από το άνοιγμα / κλείσιμο του ασυρμάτου.
- Κατά την διάρκεια εκπομπής και εξαρτάται από την ισχύ μετάδοσης.
- Κατά την αναμονή λήψης δεδομένων.
- Κατά την επεξεργασία δεδομένων από την CPU του κόμβου.

Η ιδέα της πειραματικής εφαρμογής βασίστηκε στην προσπάθεια μεταφοράς RS232 δεδομένων όπως θα δούμε παρακάτω με μία από τις πιο διαδεδομένες σε παγκόσμια κλίμακα τεχνολογίες για ασύρματα δίκτυα αισθητήρων, IEEE 802.15.4 στην ελεύθερη μπάνα των 2.4GHz. Επίσης ήταν πολύ χρήσιμη η ερευνητική και εργασιακή εμπειρία που αποκτήθηκε στα WSNs κατά την διάρκεια της παρουσίας μου στο ερευνητικό εργαστήριο [www.RFmedia.fi](http://www.RFmedia.fi) της Φιλανδίας στο 2012 και που είχε ως αποτέλεσμα μια εργασία υπο δημοσίευση σχετικά με μετρήσεις απόδοσης σε WSN τεχνολογίες όπως περιγράφεται στο επισυναπτόμενο παράρτημα.

- Στο κεφάλαιο της Μεταφοράς Δεδομένων, παρουσιάζεται και περιγράφεται η μεταφορά δεδομένων σε υπολογιστικά συστήματα. Πιο συγκεκριμένα, αναλύονται οι όροι παράλληλη και σειριακή μεταφορά δεδομένων, ασύγχρονη μεταφορά σειριακών δεδομένων και τέλος, το πρότυπο RS232.
- Στο επόμενο κεφάλαιο παρουσιάζεται το ολοκληρωμένο κύκλωμα CC2430 SoC για το οποίο αναπτύχθηκε κώδικας γλώσσας C και περιγράφονται τα κυριότερα χαρακτηριστικά του.
- Μετά από την περιγραφή του υλικού (Hardware) της εφαρμογής, ακολουθεί περιγραφή της ασύρματης στοίβας που εκτελείται από το SoC, SimpliciTI wireless stack για δίκτυα αισθητήρων.
- Στο προτελευταίο κεφάλαιο αναλύεται η πλατφόρμα ανάπτυξης που χρησιμοποιήθηκε για τους πειραματικούς κόμβους της εφαρμογής και οι τρόποι συνδεσμολογίας αυτών.
- Στα συμπεράσματα, περιγράφονται τα προβλήματα που παρουσιάστηκαν κατά την διάρκεια της εργασίας και οι περιορισμοί που υπήρξαν κατά τη διάρκεια των πειραματικών δοκιμών. Επίσης προτείνονται μερικές λύσεις για την επίλυση των περιορισμών που εμφανίστηκαν.
- Τέλος, έχει επισυναφθεί ως παράρτημα, μια υπο δημοσίευση εργασία (Draft Paper) σχετικά με μετρήσεις τεχνολογιών ασύρματων δικτύων αισθητήρων με τις ανάλογες ασύρματες στοίβες που υποστηρίζονται. Πιο συγκεκριμένα αναπτύχθηκε κώδικας για τις πλατφόρμες: CC2510 με SimpliciTI, CC2430 με TI-MAC και CC2540 με TI-BLE για την μέτρηση της πραγματικής ταχύτητας μεταφοράς δεδομένων στο επίπεδο εφαρμογής (Application Layer benchmarking) όπως επίσης και την κατανάλωση ενέργειας (Power consumption measurements). Επίσης στο παράρτημα 2 έχει επισυναφθεί ο κώδικας που χρησιμοποιήθηκε στο C8051F020-DK αναπτυξιακό.



## ΜΕΤΑΦΟΡΑ ΔΕΔΟΜΕΝΩΝ

Ένα χαρακτηριστικό ενός υπολογιστικού συστήματος είναι η επικοινωνία δηλαδή η μεταφορά ή ανταλλαγή δεδομένων μεταξύ των τμημάτων του. Η ψηφιακή επικοινωνία μπορεί να γίνει με μια από τις δυο μορφές: παράλληλα ή σειριακά.

Στην παράλληλη επικοινωνία τα σήματα ταξιδεύουν στο διάδρομο δεδομένων ο οποίος αποτελείται από πολλές γραμμές όπου η κάθε μια από αυτές μεταδίδει κι από ένα bit της πληροφορίας, μεταδίδοντας έτσι κάθε φορά ένα ολόκληρο byte. Δηλαδή στην παράλληλη επικοινωνία όλα τα bits μιας λέξης χαρακτήρα (byte) ή και παραπάνω (word, double word) μεταδίδονται ταυτόχρονα προς τον αποδέκτη.

Στη σειριακή επικοινωνία μεταδίδεται ένα bit πληροφορίας κάθε φορά, στη σειρά μέσα στον αγωγό μεταφοράς των δεδομένων. Στην απλούστερη περίπτωση τέτοιας επικοινωνίας χρειαζόμαστε τρεις συνολικά αγωγούς, έναν για την αποστολή δεδομένων, έναν για τη λήψη και ένα που θα βρίσκεται στο δυναμικό αναφοράς (Ground) των μεταδιδόμενων σημάτων.

### Πλεονεκτήματα και μειονεκτήματα παράλληλης και σειριακής επικοινωνίας.

Το πλεονέκτημα της παράλληλης μεταφοράς δεδομένων εντοπίζεται στην ταχύτητα μετάδοσης. Είναι προφανές ότι με τον τρόπο αυτόν η μετάδοση γίνεται πολύ γρήγορα, αφού όλα τα bits μιας ψηφιολέξης (byte) μεταφέρονται ταυτόχρονα από τον πομπό στον δέκτη. Ουσιαστικά, ο ρυθμός μετάδοσης περιορίζεται από την ταχύτητα με την οποία λειτουργούν τα κυκλώματα εισόδου / εξόδου του πομπού και του δέκτη.

Το κυριότερο πλεονέκτημα της σειριακής επικοινωνίας είναι ο μικρότερος αριθμός καλωδίων διασύνδεσης που απαιτείται σε σχέση με την παράλληλη επικοινωνία. Αυτό κάνει την εγκατάσταση φθηνότερη όταν οι αποστάσεις είναι μεγάλες.

Επιπλέον, τα πρωτόκολλα επικοινωνίας που χρησιμοποιούνται στη σειριακή επικοινωνία επιτρέπουν μεγάλες στάθμες σημάτων σε σχέση με τα πρωτόκολλα της παράλληλης επικοινωνίας, οπότε οι απώλειες του σήματος δημιουργούν μικρότερο πρόβλημα και η μετάδοση σε μεγάλη απόσταση είναι εφικτή.

Επίσης, στην παράλληλη επικοινωνία, επειδή έχουμε πολλούς αγωγούς που μεταδίδουν σήματα ταυτόχρονα, έχουμε παρεμβολή μεταξύ τους κάνοντας την παράλληλη επικοινωνία ακατάλληλη σε μακρινές αποστάσεις.

Για τους παραπάνω λόγους η παράλληλη επικοινωνία χρησιμοποιείται κυρίως στη διασύνδεση συσκευών σε μικρές αποστάσεις, όταν υπάρχει ανάγκη για υψηλούς ρυθμούς μετάδοσης της πληροφορίας. Έτσι, συσκευές όπως οι εκτυπωτές και οι οπτικοί σαρωτές, που απαιτούν ταχεία μεταφοράς σημαντικής ποσότητας πληροφορίας, διασυνδέονται μέσω παράλληλης επικοινωνίας με τον ηλεκτρονικό υπολογιστή.

### **Σειριακή Επικοινωνία**

Ο πιο πολύ διαδεδομένος τρόπος μετάδοσης της πληροφορίας μεταξύ των υπολογιστικών συστημάτων, είναι η σειριακή επικοινωνία για λόγω χαμηλού κόστους και ο χαμηλός θόρυβος για μεγάλες αποστάσεις.

Επειδή τα διάφορα ολοκληρωμένα κυκλώματα που απαρτίζουν έναν ηλεκτρονικό υπολογιστή ανταλλάσσουν πληροφορίες σε παράλληλη μορφή, για να είναι ο Η/Υ συμβατός και με σειριακή επικοινωνία αυτά τα σήματα πρέπει να μετατραπούν σε σειριακή μορφή. Τη λειτουργία αυτή αναλαμβάνει ένα κύκλωμα που ονομάζεται UART (Universal Asynchronous Receiver/Transmitter) το οποίο υπάρχει σε μορφή ολοκληρωμένου επάνω στη μητρική πλακέτα. Πλέον με την ραγδαία ανάπτυξη της τεχνολογίας κυκλώματα USART έχουν ενσωματωθεί στους μικροελεγκτές για την ευκολότερη επικοινωνία αυτών με τις διάφορες συσκευές αλλά και με άλλους μικροελεγκτές. Ο κλασικός 8051 μικροελεγκτής περιέχει ένα κύκλωμα USART όπως θα δούμε στα παρακάτω κεφάλαια.

Στην ασύρματη μετάδοση πληροφορίας, χρησιμοποιείται η σειριακή επικοινωνία, όπως για παράδειγμα οι διατάξεις επικοινωνίας μέσω υπέρυθρης ακτινοβολίας LED ή LASER. Η σειριακή μετάδοση είναι πιο κατάλληλη για χρήση με μικροελεγκτές, για το λόγο ότι οι διάφορες περιφερειακές διατάξεις, όπως μετατροπείς A/D, μνήμες κατασκευάζονται με λιγότερους ακροδέκτες όταν επικοινωνούν σειριακά. Εξάλλου, μερικά συστήματα μικροελεγκτών έχουν ενσωματωμένες θύρες σειριακής διασύνδεσης με το εξωτερικό περιβάλλον, κάτι που καθιστά απλή τη σειριακή διασύνδεση τους με περιφερειακές συσκευές.

Στη σειριακή επικοινωνία, εάν τα δεδομένα μπορούν να εκπέμπονται και να λαμβάνονται τότε λέμε ότι έχουμε ότι έχουμε duplex επικοινωνία. Αυτό είναι σε αντίθεση με την simplex επικοινωνία όπως στους εκτυπωτές που το υπολογιστικό σύστημα μόνο εκπέμπει. Duplex επικοινωνίες μπορεί να είναι half ή full duplex που εξαρτάται εάν η μεταφορά δεδομένων μπορεί ή όχι να γίνεται ταυτόχρονα. Εάν τα δεδομένα εκπέμπονται σε μια κατεύθυνση τη φορά, τότε

αναφέρεται σαν half duplex. Εάν τα δεδομένα εκπέμπονται και στις δυο κατευθύνσεις την ίδια στιγμή, έχουμε full duplex.

Για την σειριακή μετάδοση σε ακόμη μεγαλύτερες αποστάσεις π.χ κατά μήκος του τηλεφωνικού δικτύου, παρεμβάλλονται ειδικές συσκευές που ονομάζονται modems για κωδικοποίηση (μετατροπή από 0 και 1 σε ακουστικούς τόνους) και αποκωδικοποίηση (μετατροπή των ακουστικών τόνων σε 0 και 1).-

Για να είναι εφικτή η σειριακή επικοινωνία μεταξύ δυο υπολογιστικών συστημάτων, πρέπει να ακολουθούν και τα δυο ένα καθορισμένο σύνολο κανόνων, που ονομάζονται πρωτόκολλα επικοινωνίας. Τέτοια πρωτόκολλα επικοινωνίας είναι το RS232, το I2C, το SPI, το Ethernet κ.ά. Η σειριακή επικοινωνία ανάλογα αν εκπέμπονται σήματα συγχρονισμού μεταξύ πομπού και δέκτη διακρίνεται σε ασύγχρονη ή σύγχρονη σειριακή επικοινωνία.

### **Ασύγχρονη Σειριακή Επικοινωνία**

Η ασύγχρονη σειριακή επικοινωνία είναι επικεντρωμένη στην μετάδοση χαρακτήρων δηλαδή δεδομένων με συγκεκριμένο πλάτος bits και στην οποία δεν εκπέμπονται σήματα συγχρονισμού.

Ένας χαρακτήρας δεδομένων μετατρέπεται σε ψηφιολέξη μέσω του κώδικα ASCII και η ακολουθία bits που αντιστοιχεί σε αυτόν τον χαρακτήρα εμφανίζεται στη γραμμή μετάδοσης. Ο δέκτης πρέπει να είναι σε θέση να αναγνωρίζει ότι έφτασε ένας χαρακτήρας και να αναγνωρίζει τα bits του χαρακτήρα με τη σωστή σειρά και χωρίς απώλειες.

Για τον παραπάνω σκοπό, τα bits της ασύγχρονης σειριακής μετάδοσης οργανώνονται σε πλαίσια των εννέα ως δώδεκα bits συνολικά, τα οποία περιέχουν κάποια bits έναρξης και λήξης. Το πρώτο bit κάθε πλαισίου είναι το λεγόμενο start bit, το οποίο αντιστοιχεί σε λογικό μηδέν. Ακολουθεί η σειρά των ψηφίων του χαρακτήρα που αποστέλλεται. Μετά από τα bits του χαρακτήρα, ακολουθεί ένα bit άρτιας ή περιττής ισοτιμίας (parity), το οποίο ενεργοποιεί μια διαδικασία ελέγχου σφαλμάτων, για να ανιχνεύσει τυχόν λάθη που συνέβησαν κατά τη μετάδοση. Το πλαίσιο κλείνει με ένα ή δυο stop bits, που υποδηλώνουν το τέλος του χαρακτήρα και την κατάσταση αναμονής για τον επόμενο. Η λογική κατάσταση των ψηφίων λήξης (stop bits) είναι ένα.

Όταν δεν μεταδίδεται κάποιος χαρακτήρας, τότε λέμε ότι η σύνδεση είναι ανενεργή, οπότε η γραμμή ευρίσκεται σε λογικό ένα. Η κατάσταση αυτή ονομάζεται «συνθήκη mark». Όταν μεταδοθεί το bit έναρξης (start bit) και φθάσει στο δέκτη, ο δέκτης καταλαβαίνει ότι ακολουθούν τα bits του χαρακτήρα που αποστέλλεται οπότε ενεργοποιεί το σύστημα χρονισμού και διαβάζει με τη σειρά τα επόμενα bits μέχρι τα bits λήξης (stop bits). Στη συνέχεια τίθεται στην κατάσταση αναμονής.

Είναι φανερό ότι η διάρκεια του κάθε εκπεμπόμενου bit στην ασύγχρονη σειριακή μετάδοση πρέπει να είναι αυστηρά η ίδια, ώστε να μπορεί ο δέκτης, με κάποιο σύστημα συγχρονισμού, να διακρίνει τα bits μεταξύ τους. Συνεπώς ο πομπός και ο δέκτης πρέπει να συμφωνούν ως προς την ταχύτητα της σειριακής μετάδοσης των bits. Η ταχύτητα αυτή ορίζει το λεγόμενο «ρυθμό μετάδοσης» που μετριέται σε bits ανά δευτερόλεπτο (bits/sec ή bps).

## **Το Πρότυπο RS-232C**

Η σειριακή θύρα συνδέεται με εξωτερικές συσκευές μέσω ενός συνδετήρα DE-9M των 9 ακροδεκτών ή DB-25 των 25 ακροδεκτών σε παλαιότερους υπολογιστές και περιφερειακά. Οι τάσεις και εν γένει τα ηλεκτρικά χαρακτηριστικά της σειριακής θύρας περιέχονται στο σειριακό πρότυπο επικοινωνίας RS-232C της Ένωσης EIA.

Το πρότυπο RS-232C επιτρέπει την ασύγχρονη επικοινωνία ανάμεσα σε δυο συσκευές. Εάν επιθυμούμε να συνδέσουμε περισσότερες από δυο συσκευές σε ένα υπολογιστή, χρειαζόμαστε περισσότερες από μια σειριακές θύρες.

Το πρότυπο RS-232C χρησιμοποιεί αρνητική ψηφιακή λογική και μεγάλες στάθμες ώστε να επιτρέπει τη διάδοση του σήματος σε μεγάλες αποστάσεις χωρίς απώλειες. Αυτά έχουν σαν αποτέλεσμα οι τάσεις του πρωτοκόλλου RS-232C να μην είναι συμβατές με τις στάθμες TTL.

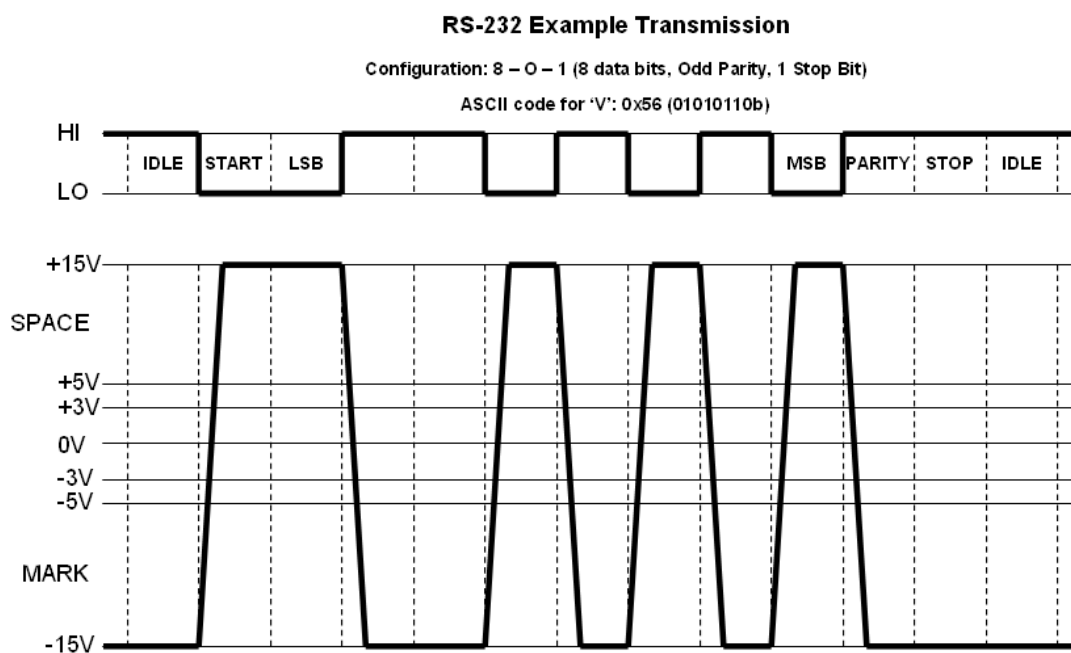
Τα επίπεδα τάσεων του πρωτοκόλλου RS-232C, είναι τα εξής:

- Το λογικό 0, που λέγεται SPACE, βρίσκεται μεταξύ +3 και +25V (πρακτικά από +5 έως +15V)
- Το λογικό 1, που λέγεται MARK, βρίσκεται μεταξύ -3 και -25V (πρακτικά από -5 έως -15V)
- Η περιοχή από -3V έως +3V δεν αντιπροσωπεύει καθορισμένη λογική στάθμη.

- Κανέναν από τους ακροδέκτες της σειριακής θύρας δεν μπορεί να δεχτεί δυναμικό μεγαλύτερο από 25V σε σχέση με τη γη.
- Το μέγιστο ρεύμα δεν μπορεί να ξεπερνά τα 500mA

Με χρήση του πρωτοκόλλου RS-232C, ένα τερματικό χαρακτήρων μπορεί να αποστείλει μέσω μιας γραμμής επικοινωνίας δεδομένα, σύμφωνα με τους κανόνες της ασύγχρονης σειριακής μετάδοσης που περιγράψαμε στην προηγούμενη ενότητα. Όταν η γραμμή είναι ανενεργή, ευρίσκεται σε συνθήκη MARK, δηλαδή -12V περίπου, που αντιστοιχεί στο λογικό 1. Η γραμμή ενεργοποιείται με τη συνθήκη SPACE (λογικό 0 ή +12V). Ακολουθεί η μετάδοση επτά ή οκτώ bits για το υποστελλόμενο χαρακτήρα, ένα προαιρετικό bit άρτιας ή περιττής ισοτιμίας και ένα ή δυο stop bits (συνθήκη MARK), που σηματοδοτούν το τέλος του χαρακτήρα.

Στο **Σχήμα 1** φαίνονται οι λογικές και RS232 στάθμες του χαρακτήρα 'V' με χαρακτηριστικά: 8bits δεδομένων, περιττή ισοτιμία και 1 bit λήξης.



Σχήμα 1

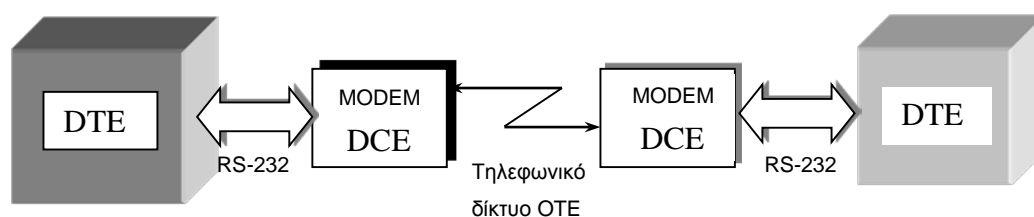
## Τύποι Ασύγχρονων Σειριακών Συσκευών

Το πρότυπο RS-232 καθορίζει δύο τύπους συσκευών. Ο πρώτος τύπος συσκευής ονομάζεται *Τερματική Συσκευή Δεδομένων (Data Terminal Equipment ή DTE)*. Ο δεύτερος τύπος συσκευής ονομάζεται *Συσκευή Επικοινωνίας Δεδομένων (Data Communications Equipment ή DCE)*.

Κάθε προσωπικός υπολογιστής και σχεδόν κάθε τερματικός σταθμός που διαθέτει θύρα RS-232 είναι συσκευή DTE.

Για τη διασύνδεση δύο τερματικών σταθμών σε μεγάλες αποστάσεις πριν την ευρεία διάδοση της xDSL τεχνολογίας στο νοικοκυριό χρησιμοποιούνταν συσκευές επικοινωνίας τύπου modem σειριακής ασύγχρονης επικοινωνίας για την σύνδεση στο Internet όπως εικονίζεται στο **Σχήμα 2**. Τα modem αναφέρονται ως *συσκευές DCE*.

Εκτός από τα modem, στη σειριακή θύρα ενός υπολογιστή μπορεί να συνδεθούν κι άλλες συσκευές που δεν προορίζονται για επικοινωνία, αλλά για μετρήσεις, έλεγχο κυκλωμάτων κ.λπ.



Σχήμα 2

Πολλά πολύμετρα, συσκευές δικτύου, UPS, ελεγκτές (controllers), αισθητήρες, ακόμη και οικιακές ηλεκτρονικές συσκευές όπως video, δορυφορικοί δέκτες κλπ. έρχονται εξοπλισμένα με σειριακή θύρα RS-232. Οι συσκευές αυτές συνήθως είναι διαμορφωμένες σαν DCE και το καλώδιο που τις συνδέει με τον υπολογιστή καθώς και τα σήματα ελέγχου που ανταλλάσσουν μαζί του είναι ίδια με αυτά ενός modem.

Είναι σημαντικό να αναφερθεί ότι οι ταχύτητες ανάμεσα σε μία συσκευή DTE και σε μία συσκευή DCE είναι γενικά διαφορετικές από τις ταχύτητες επικοινωνίας ανάμεσα σε δύο συσκευές DCE. Αυτό συμβαίνει όχι μόνο διότι στη δεύτερη περίπτωση παρεμβάλλεται το τηλεφωνικό δίκτυο, αλλά και διότι τα ίδια τα κυκλώματα των συσκευών DCE λειτουργούν σε διαφορετική ταχύτητα προς την κατεύθυνση επικοινωνίας με συσκευές DCE, παρά προς την κατεύθυνση επικοινωνίας με συσκευές DTE (). Έτσι είναι δυνατό να συνδέσουμε ένα modem στη σειριακή θύρα COM1 του υπολογιστή μας, ρυθμίζοντας την ταχύτητα της σύνδεσης στη μέγιστη τιμή των 115.2 kbps, ενώ το modem θα συνδέεται με κάποιο άλλο modem στην πολύ μικρότερη

ταχύτητα των 28.8 kbps. Αυτή η δεύτερη ταχύτητα αποτελεί χαρακτηριστικό γνώρισμα της συσκευής DCE που χρησιμοποιούμε.

### Οι Ακροδέκτες της Σειριακής Θύρας και οι Λειτουργίες τους

Ανάμεσα στους ακροδέκτες της σειριακής θύρας διακρίνουμε τους *ακροδέκτες ή γραμμές δεδομένων (data lines)* και τους *ακροδέκτες ή γραμμές ελέγχου (control lines)*. Οι σπουδαιότεροι ακροδέκτες είναι αυτοί που μεταφέρουν τα δεδομένα εκπομπής και λήψης, δηλαδή την πληροφορία προς την μία ή την άλλη κατεύθυνση. Όλοι οι υπόλοιποι ακροδέκτες είναι βοηθητικοί αλλά απαραίτητοι για την επικοινωνία ενός DTE με ένα DCE.

Οι γραμμές δεδομένων ονομάζονται TXD και RXD αντιστοιχούν στους ακροδέκτες 2 και 3 στον DE-9M σύνδεσμο όπως φαίνεται στον **πίνακα 1**. Η γραμμή TXD προορίζεται για την εκπομπή των σειριακών δεδομένων, η γραμμή RXD για τη λήψη των δεδομένων, ενώ η γραμμή GND είναι ο αγωγός αναφοράς των τάσεων που διαδίδονται στις πρώτες δύο γραμμές.

Οι σπουδαιότερες από τις γραμμές ελέγχου είναι οι RTS, CTS, DSR και DTR. Όταν ο υπολογιστής (DTE) θέλει να στείλει δεδομένα σε μια συσκευή DCE, ενεργοποιεί τη γραμμή RTS (*Request To Send*) θέτοντας στη γραμμή τη συνθήκη SPACE. Εάν η συσκευή DCE έχει χώρο για να δεχθεί δεδομένα, τότε απαντά ενεργοποιώντας τη γραμμή CTS (*Clear To Send*).

Αντίστοιχα, όταν το modem θέλει να στείλει δεδομένα, πληροφορεί τη θύρα UART ενεργοποιώντας τη γραμμή DSR (*Data Set Ready*). Εάν ο υπολογιστής από την πλευρά του είναι έτοιμος να λάβει δεδομένα, απαντά θέτοντας την κατάσταση SPACE στον ακροδέκτη DTR (*Data Terminal Ready*). Εάν, πάλι, δεν είναι έτοιμος να λάβει δεδομένα, τότε θέτει συνθήκη MARK στη γραμμή DTR.

Από τα παραπάνω γίνεται φανερό ότι από τη σκοπιά του υπολογιστή οι ακροδέκτες DTR και RTS προορίζονται για ΕΞΟΔΟ δεδομένων, ενώ οι γραμμές CTS και DSR προορίζονται για ΕΙΣΟΔΟ δεδομένων.

Είναι προφανές ότι με την χρήση των γραμμών RTS/CTS και DTR/DSR στην ουσία επιτυγχάνεται ο έλεγχος της ροής των εκπεμπόμενων - λαμβανόμενων δεδομένων. Αυτό δεν είναι απαραίτητα αναγκαίο και εξαρτάται από την ταχύτητα των καναλιών μετάδοσης δεδομένων ή/και από τις απαιτήσεις της εφαρμογής για αξιόπιστη ανταλλαγή δεδομένων. Αυτό

συμβαίνει ότι όταν έχουμε π.χ. ασύρματη ζεύξη θα πρέπει να με κάποιο τρόπο να υπάρχει έλεγχος της ταχύτητας ροής από τα μεταφερόμενα συστήματα στα συστήματα μεταφοράς των δεδομένων στην ασύρματη ζεύξη για την επίτευξη αξιόπιστης επικοινωνίας με όσο το δυνατόν λιγότερα λάθη.

<b>Ακροδέκτες συνδέσμου DE-9M</b>	<b>Κωδική ονομασία</b>	<b>Όνομα</b>
3	TXD	Transmit Data
2	RXD	Receive Data
5	GND	Common Ground
7	RTS	Request To Send
8	CTS	Clear To Send
6	DSR	Data Set Ready
4	DTR	Data Terminal Ready
1	CD	Carrier Detect
9	RI	Ring Indicator

*Πίνακας 1*



## Το CC2430 SoC

Υπάρχουν στην αγορά διάφοροι τύποι μικροελεγκτών (MCU ICs) και πομποδεκτών (RF TRANSCEIVER ICs) με τα οποία τα ασύρματα δίκτυα αισθητήρων υλοποιούνται. Είναι κατασκευασμένα είτε σε ξεχωριστά ολοκληρωμένα κυκλώματα (ICs) είτε σε ένα μόνο ολοκληρωμένο (System On Chip - SoC). Η πτυχιακή είναι βασισμένη στο CC2430 SoC ολοκληρωμένο της Texas Instruments (TI), μια πολύ μεγάλη εταιρία ηλεκτρονικών της Αμερικής, αν και ο πραγματικός σχεδιαστής ήταν η Chipcon μια μικρή εταιρία στην Νορβηγία όπου και αγοράστηκε από την TI.

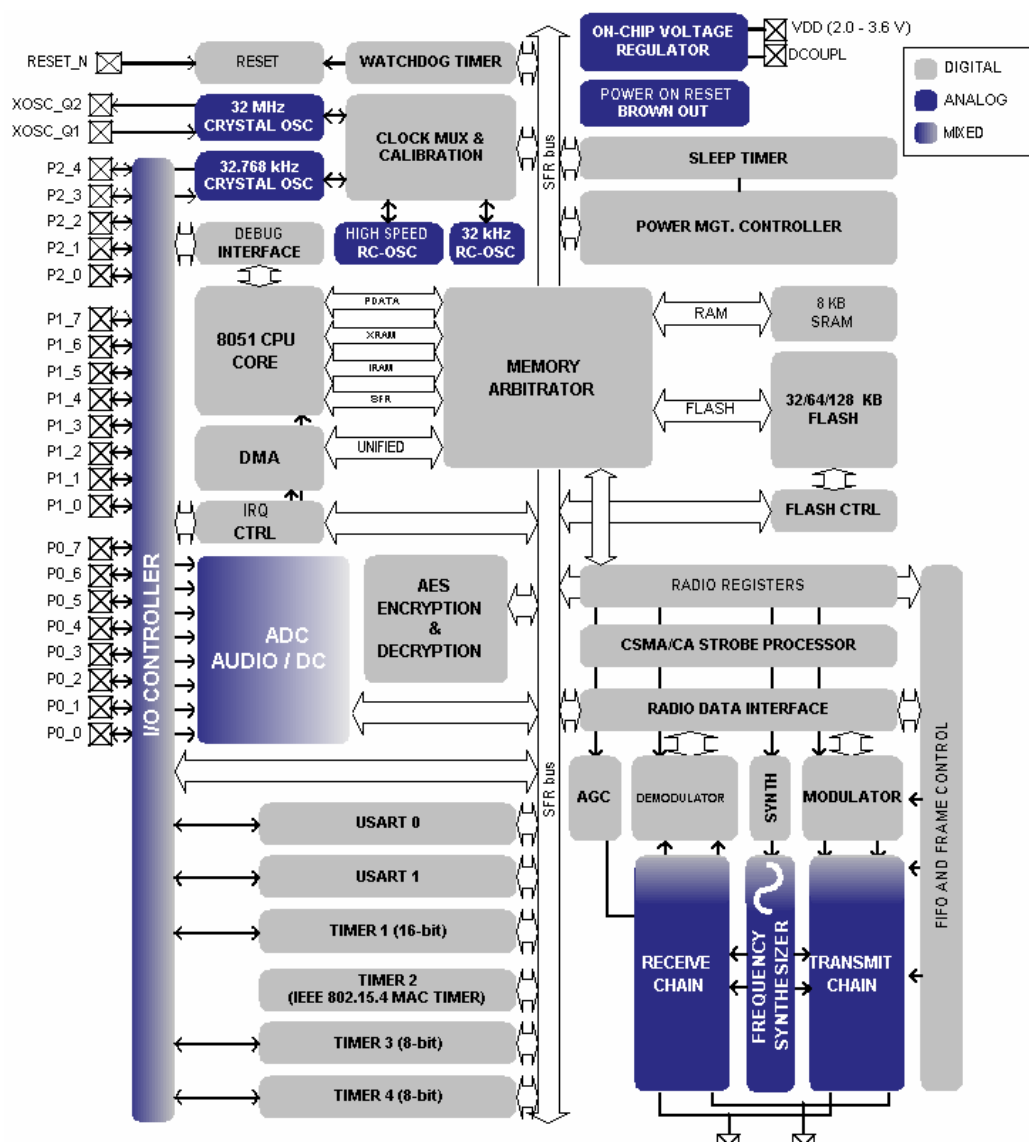
Μερικά πλεονεκτήματα της SoC υλοποίησης είναι το μικρότερο κόστος μιας και χρειάζεται ένα μόνο IC για να κατασκευάσουμε ένα κόμβο ασύρματου δικτύου σε αντίθεση με την υλοποίηση με ξεχωριστό RF IC από τον μικροελεγκτή (MCU) και απαιτείται λιγότερος χώρος στο τυπωμένο κύκλωμα (PCB). Το CC2430 κατασκευάζεται σε μόνο σε QFN κέλυφος (**Φωτογραφία 1**) και βοηθάει ακόμα περισσότερο μιας και καταλαμβάνει 7mm x 7mm. Στην παραδοσιακή υλοποίηση η ταχύτητα μετάδοσης από MCU σε RF IC περιορίζεται από την ταχύτητα λειτουργίας του σειριακού πρωτοκόλλου μεταφοράς των δεδομένων. Σε αυτή την περίπτωση χρειάζεται επεξεργαστική ισχύς από τον MCU σε κάθε μεταφερόμενο byte από το σειριακό πρότυπο επικοινωνίας SPI που συνήθως χρησιμοποιεί ο μικροελεγκτής σε σχέση με την SoC υλοποίηση που η CPU μπορεί να προσπελάσει τους καταχωρητές άμεσα, εκτός και αν υποστηρίζεται η μεταφορά δεδομένων με χρήση DMA αλλά ακόμα και DMA να υπάρχει η κατανάλωση ισχύος στην περίπτωση των 2 ICs είναι περισσότερη συνήθως αφού τα δεδομένα μεταφέρονται με τον εξωτερικό παραδοσιακό τρόπο αγωγών μεταφοράς στο PCB.



Φωτογραφία 1

## Μπλοκ Διάγραμμα & Χαρακτηριστικά

Το CC2430 περιέχει τα παρακάτω στοιχεία και περιφερειακά, όπως παρουσιάζονται στο μπλόκ διάγραμμα της **Εικόνα 2**.



Εικόνα 2

Το CC2430 περιληπτικά έχει τις παρακάτω δυνατότητες και χαρακτηριστικά:

- Βελτιωμένη CPU 8051 συμβατός με τυπική βελτίωση X8 από την απόδοση του κλασικού 8051 με δύο 16bit δείκτες μνήμης DPTRs.
- Ενσωματωμένη μνήμη μέχρι και 128KB τεχνολογίας Flash με υποστήριξη περιοχών προστατευόμενης ανάγνωσης/εγγραφής.

- Ενσωματωμένη μνήμη RAM 8KBytes συνολικά. 4KB με συγκράτηση δεδομένων σε όλες τις καταστάσεις “ύπνωσης” και ακόμα 4KB όπου τα αποθηκευμένα δεδομένα συγκρατούνται στις καταστάσεις PM0 και PM1 (εξοικονόμησης ενέργειας).
- Συνεπεξεργαστής κρυπτογραφίας AES 128bits.
- Ευέλικτος DMA ελεγκτής 5 καναλιών με πολλές δυνατότητες και επιλογές τύπου λειτουργίας.
- Reset κατά την έναρξη της τροφοδοσίας και ελεγκτής τάσης τροφοδοσίας για αυτόματο reset σε περιπτώσεις αστάθειας της τάσης.
- Ενσωματωμένος μετατροπέας αναλογικού σήματος σε ψηφιακό των 12bit (ADC) με έως και 8 κανάλια εισόδου.
- Παραμετροποιήσιμος χρονιστής Watchdog.
- Ρολόι πραγματικού χρόνου (RTC) με ταλαντωτή κρύσταλλου των 32.768 kHz
- 4 εσωτερικούς χρονιστές: Ένας 16bit γενικού σκοπού 2 ακόμα των 8bit επίσης γενικού σκοπού και τέλος χρονιστής ειδικής λειτουργίας IEEE 802.15.4 MAC
- 2 παραμετροποίηση USARTs για ασύγχρονη/σύγχρονη σειριακή επικοινωνία ακόμα και σε SPI.
- 21 ρυθμιζόμενοι ακροδέκτες γενικού σκοπού εισόδου / εξόδου δεδομένων
- Γεννήτρια δημιουργίας πραγματικά τυχαίων αριθμών
- 4 ευέλικτες καταστάσεις λειτουργίας για ελάττωση της κατανάλωσης ισχύος τους συστήματος
- Αφύπνιση λειτουργίας από εξωτερική διακοπή ή από γεγονός του μετρητή (Counter event)
- Πλήρως στατική CMOS σχεδίαση για χαμηλή κατανάλωση

#### Χαρακτηριστικά RF:

- Ενσωματωμένος πομποδέκτης στην ISM μπάντα των 2.4GHz συμβατός με το πρωτόκολλο IEEE 802.15.4.
- Άριστη ευαισθησία δέκτη με αντοχή στις παρεμβολές με τυπική ευαισθησία λήψης τα -92dBm.
- 250 kbps ταχύτητα μεταφοράς δεδομένων με τεχνική εξάπλωση φάσματος (DSSS)
- Αυτόματη δημιουργία λέξεων (word) συγχρονισμού.
- Αυτόματη δημιουργία προοιμίου μετάδοσης (Preamble).
- Αυτόματη τοποθέτηση και αναγνώριση λέξεων συγχρονισμού.

- Αυτόματος υπολογισμός CRC-16 και έλεγχος αυτού για την ανίχνευση σφαλμάτων.
- Έλεγχος διαθεσιμότητας καναλιού (CCA)
- Ανίχνευση λαμβανόμενης ισχύς (Energy Detection-ED), υπολογισμού δείκτη έντασης λαμβανόμενου σήματος (Received Signal Strength Indicator - RSSI) και δείκτη ποιότητας ασύρματης ζεύξης (Link Quality Indication - LQI).
- Συνεπεξεργαστής CSMA/CA για αποφυγή συγκρούσεων.
- Προσχεδιασμένα παραδείγματα ανάπτυξης (reference designs) από την TI συμμορφωμένα με παγκόσμιους κανονισμούς όπου αναφέρονται στα standards: ETSI EN 300 328 και EN 300 440 class 2 (Europe), FCC CFR47 Part 15 (US) και ARIB STD-T66 (Japan). Εκπομπή στα 2480MHz με το FCC standard υποστηρίζεται αλλά απαιτείται η εκπομπή με περιορισμένο κύκλο λειτουργίας (Duty cycle) ή/και περιορισμός της εκπεμπόμενης ισχύος

Άλλα χαρακτηριστικά:

- Επιτρεπτό εύρος τάσης λειτουργίας του CC2430: Ελάχιστη τάση 2.0 Volts με μέγιστη τα 3.6Volts.
- Τυπική κατανάλωση ισχύος με 3V τάση τροφοδοσίας και σε περιβάλλον 25°C είναι κατά εκπομπή 0dbm είναι  $26,9\text{mA} = 80,7 \text{ mW}$  και για λήψη  $26,7\text{mA} = 80,1 \text{ mW}$ .
- Θερμοκρασίες λειτουργίας από -40 έως 85 βαθμούς κελσίου.

## Ταλαντωτές

Η καρδιά του CC2430 αποτελείται από τους ταλαντωτές: Α) Τον εσωτερικής σταθεράς RC 16MHz συχνότητας ταλαντωτή που είναι προεπιλεγμένη επιλογή (default) του καταχωρητή CLKCON «ρολογιού λειτουργίας» (System Clock) κατά την εκκίνηση του μικροελεγκτή. Β) Τον ταλαντωτή παραγωγής συχνότητας 32MHz με εξωτερικό κρύσταλλο ακριβείας για την λειτουργία της CPU στην μέγιστη δυνατή ταχύτητα λειτουργίας με μέγιστη δυνατή σταθερότητα λειτουργίας μιας και οι ταλαντωτές τύπου XTAL έχουν την μικρότερη απόκλιση από την επιθυμητή ονομαστική συχνότητα κατασκευής. Στην περίπτωση που θα χρησιμοποιηθεί από την εφαρμογή ο RF πομποδέκτης πρέπει να ενεργοποιηθεί ο ταλαντωτής των 32MHz που απαιτείται από το PLL για την δημιουργία της φέρουσας υψηλής συχνότητας των 2.4GHz. Ο MCU μπορεί να λειτουργεί είτε στα 32MHz είτε στα 16MHz για λιγότερη κατανάλωση ισχύος κατά την λειτουργία του (CPU active). Επίσης προαιρετικά μπορεί να χρησιμοποιούνται ως System Clock οι ταλαντωτές πολύ

μικρής κατανάλωσης 32.768kHz RC ή εξωτερικού κρυστάλλου XTAL 32.768kHz σε καταστάσεις χαμηλής επεξεργαστικής ανάγκης για σημαντική μείωση κατανάλωσης ισχύος. Στο παρακάτω πίνακα παρουσιάζονται οι καταστάσεις λειτουργίας – εξοικονόμησης ενέργειας με τους ταλαντωτές να απενεργοποιούνται.

## CPU

Η CPU είναι συμβατή με την αρχιτεκτονική 8051 των 8bit που σχεδίασε η intel πριν από πολλά χρόνια αλλά ακόμα χρησιμοποιείται και κατασκευάζεται από την βιομηχανία λόγω της απλότητας σχεδίασης του, την ευκολία προγραμματισμού του σε assembly αλλά και του χαμηλού κόστους κατασκευής. Σε κάποιες περιπτώσεις φαίνεται η παλαιότητα της αρχιτεκτονικής λόγω της περιορισμένης επεξεργαστικής ισχύς που προσφέρει στον προγραμματιστή και γίνεται αντιληπτό κυρίως σε εφαρμογές με πολλές αριθμητικές πράξεις ιδιαίτερα με τιμές μεγαλύτερες των 8bit. Βέβαια η συγκεκριμένη υλοποίηση του 8051 (8052 στην πραγματικότητα) είναι πολύ βελτιωμένη όπως θα αναλύσουμε παρακάτω σε σχέση με αρχική σχεδίαση από την intel. Η επεξεργαστική ισχύς της CPU έχει βελτιωθεί αφού χρειάζεται 1 CLK για την φόρτωση και εκτέλεση αρκετών εντολών 1 Byte και σημαίνει ότι εκτελούνται 12 φορές γρηγορότερα σε σχέση με την αρχική σχεδίαση του 8051. Επίσης εμπεριέχει 2 DPTR καταχωρητές έτσι μπορεί να γίνει αντιγραφή δεδομένων γρηγορότερα από Flash σε Xdata ή και από Xdata σε Xdata αν και η χρήση DMA είναι καλύτερη για μεταφορά πολλών δεδομένων. Ο κατασκευαστής υποστηρίζει ότι η 8051 CPU του CC2430 είναι κατά 8 φορές πιο γρήγορος.

## RAM (Data, iData, Xdata)

Η CPU εμπεριέχει εσωτερική μνήμη SRAM ταχείας προσπέλασης μεγέθους 256bytes ονομαζόμενη DATA/iDATA:

- Η περιοχή 00-1Fh συνολικού μεγέθους 32bytes είναι η μνήμη των καταχωρητών γενικού σκοπού (GPRs).
- Η περιοχή 20h-2Fh συνολικού μεγέθους 16bytes είναι η μνήμη που μπορεί να γίνει επεξεργασία των δεδομένων της ανά bit και όχι ανά byte όπως συνηθίζεται.
- Η περιοχή 30h-7Fh συνολικού μεγέθους 80bytes είναι μνήμη γενικού σκοπού με δυνατότητα άμεσης προσπέλασης από τον πυρήνα ή και για την χρήση του σωρού.
- Η περιοχή 80h-FFh περιέχει διάφορους καταχωρητές ειδικού σκοπού (SFRs) που χρησιμοποιούνται για την ενεργοποίηση, παραμετροποίηση αλλά και για την

κατάσταση λειτουργίας των διάφορων περιφερειακών του MCU π.χ σειριακή (SCON, SBUF), χρονιστές (TMOD) κλπ. είτε για αποστολή/λήψη δεδομένων άμεσα όπως για παράδειγμα από τους καταχωρητές P0, P1, P2 όπου είναι οι πόρτες εισόδου (I/Os) εξόδου του CC2430.

- Τέλος η περιοχή 80h-FFh είναι μνήμη γενικού σκοπού που όμως μπορεί να προσπελασθεί ΜΟΝΟ με έμμεση διευθυνσιοδότηση γιατί αυτή η περιοχή διευθύνσεων μνήμης είναι αλληλεπικαλυπτόμενη με κάποιες από τις διευθύνσεις.

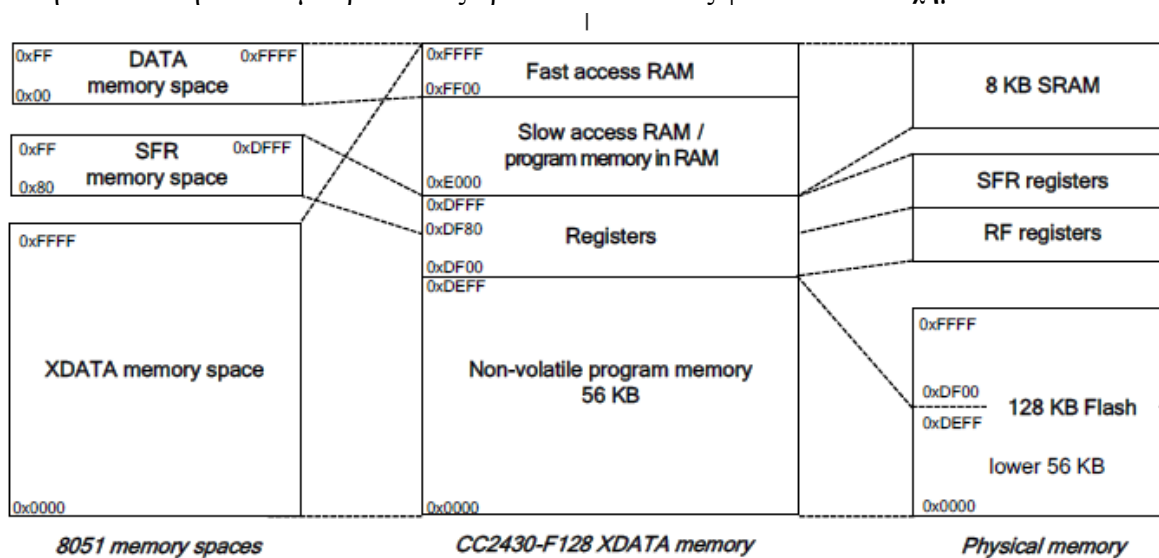
Εκτός από την εσωτερική μνήμη των 256bytes το CC2430 εμπεριέχει μνήμη SRAM συνολικού μεγέθους 8192Bytes (2 X 4KBytes) ονομαζόμενη XDATA. Η μνήμη αυτή είναι ποιο αργή και χρειάζονται 3-11 κύκλοι ρολογιού (CLK Cycles) γιατί δεν είναι διασυνδεδεμένη άμεσα με το κεντρικό δίαυλο του πυρήνα και η προσπέλαση της από τη CPU γίνεται μέσω του memory crossbar/arbitrer με έμμεση διευθυνσιοδότηση με την εντολή MOVX και των καταχωρητών R0 ή/και R1 αλλά και του ειδικού καταχωρητή MPAGE που πρέπει να έχει τα 8 περισσότερα σημαντικά bit της διεύθυνσης XDATA που πρέπει να προσπελαστεί.

## **ROM (Flash)**

Η μνήμη προγράμματος είναι η μνήμη στην οποία υπάρχει υποθηκευμένο το πρόγραμμα εκτέλεσης και φορτώνεται (Opcode fetch) από τον MCU μέσω του δείκτη μνήμης προγράμματος των 16Bit (Program Counter, PC) αν και μπορεί να προσπελάσει >64Kbyte ROM με την χρήση της τεχνικής των memory banks και Bank switching. Το πρόγραμμα εκτέλεσης υπάρχει κυρίως στην μνήμη ROM τύπου FLASH αλλά η έκδοση του CC2430 εκτός ότι υποστηρίζει 128Kbyte Flash ως μνήμη προγράμματος CODE διαιρεμένη σε 4banks των 32KB υποστηρίζει και εκτέλεση εντολών προγράμματος από την XDATA SRAM.

## Memory Crossbar - Arbitrator

Ο memory crossbar/arbitrator χρησιμοποιείται για την διασύνδεση της CPU και του ελεγκτή DMA με όλες τις μνήμες (DATA, iDATA, XDATA, CODE) ανεξάρτητα, αλλά και με όλα τα περιφερειακά μέσω του SFR διαύλου. Έτσι μπορεί να επιτευχθεί ταυτόχρονη ανεξάρτητη προσπέλαση μνημών από τον DMA και την CPU για την αύξηση των επιδόσεων. Αυτό επιτυγχάνεται γιατί αυτός ο ελεγκτής δημιουργεί έναν διάυλο διευθύνσεων μνήμης (address memory range) στην XDATA περιοχή και περιέχει όλες τις μνήμες (data, code, xdata) που κανονικά εκτός του ότι έχουν διαφορετικές διευθύνσεις είναι και διαφορετικές μνήμες. Με αυτή την τεχνική ο DMA ελεγκτής να μπορεί να τις “δεί” όλες ανεξάρτητα από την πραγματική τους φυσική διεύθυνση και να μπορεί να τις προσπελάσει όπως φαίνεται στο **Σχήμα 3**.



Σχήμα 3

## DMA

Αναλυτικότερα ο Direct Memory Controller (DMA) είναι ένα περιφερειακό τύπου ελεγκτή που επιτυγχάνει την αποσυμφόρηση της CPU για μεταφορές δεδομένων μνήμης και έτσι επιτυγχάνεται μεγάλη απόδοση του συστήματος χωρίς μεγάλη αύξηση κατανάλωσης ισχύος από την χρήση του. Αυτό γιατί δεν απασχολείται η CPU για την μεταφορά δεδομένων με την χρήση του DMA έτσι η CPU πχ. να απασχολείται με την επεξεργασία των επόμενων για μεταφορά δεδομένων όπου σε αυτόν το χρόνο το DMA απασχολείται ήδη για την μεταφορά των ήδη επεξεργασμένων δεδομένων. Ο συγκεκριμένος DMA controller του CC2430 έχει τα εξής χαρακτηριστικά:

- 5 Ανεξάρτητα κανάλια DMA για ταυτόχρονη μεταφορά δεδομένων
- 3 παραμετροποιήσιμα επίπεδα προτεραιότητας των καναλιών DMA

- 31 παραμετροποιήσιμα γεγονότα μεταφοράς
- Ανεξάρτητος έλεγχος των διευθύνσεων αποστολής και προορισμού δεδομένων.
- Δυνατότητα επιλογής μεγέθους μεταφοράς. Ένα ή πολλαπλά bytes σε σειρά (μπλοκ) όπως και δυνατότητα επαναλαμβανόμενων μεταφορών.
- Υποστηρίζει πεδίο για διαφορετικό μήκος μεταφερόμενων δεδομένων για την ρύθμιση μεταβλητού μήκους μεταφοράς.
- Μπορεί να μεταφέρει δεδομένα μήκους 16bit (Word-size) είτε 8Bit (Byte-size).

Ο DMA ελεγκτής μπορεί να προσπελάσει όλες τις διαθέσιμες μνήμες αλλά και τους περισσότερους SFRs ως XDATA διευθύνσεις και γι' αυτό μπορεί να χρησιμοποιηθεί ακόμα και π.χ στο να φορτώνει τον καταχωρητή της σειριακής επικοινωνίας (UART) με δεδομένα από την μνήμη ή/και την περιοδική μεταφορά των δεδομένων δειγματοληψίας του ενσωματωμένου ADC. Με αυτόν τον έξυπνο τρόπο μπορεί να μειωθεί ακόμα περισσότερο η κατανάλωση ισχύος του CC2430 αφού αν εαν η CPU δεν χρειάζεται να εκτελεί εντολές όταν ο DMA μεταφέρει δεδομένα τότε μπορούμε να αλλάξουμε την κατάσταση της σε “ύπωση” με τον ελεγκτή διαχείρισης λειτουργίας – ενέργειας (Power management control).

### **Power Management Control**

Ο ελεγκτής διαχείρισης λειτουργίας και κατανάλωσης ενέργειας χρησιμοποιείται για να αλλάξει την κατάσταση λειτουργίας (PMx) του CC2430. Ανάλογα την κατάσταση, κάποια υποσυστήματα π.χ. η CPU, απενεργοποιούνται όταν δεν χρειάζονται σε κάποια χρονικά διαστήματα ώστε να επιτευχθεί όσο το δυνατόν μικρότερη κατανάλωση ισχύος. Αυτό επιτυγχάνεται: με την διακοπή της τροφοδοσίας σε κάποια από τα υποσυστήματα για την μείωση της στατικής διαρροής των MOSFET transistors που απαρτίζουν αυτά τα υποσυστήματα, με τον έλεγχο της διανομής των σημάτων ρολογιού στα υποσυστήματα και τέλος με την ολική απενεργοποίηση των ταλαντωτών που δημιουργούν τα σήματα ρολογιού.



Ο ελεγκτής διαχείρισης λειτουργίας έχει τις ακόλουθες καταστάσεις όπως παρουσιάζεται στον

**Πίνακας 2.**

Power Mode	High-frequency oscillator	Low-frequency oscillator	Voltage regulator (digital)
Configuration	A None	A None	
	B 32 MHz XOSC	B 32.753 kHz RCOSC	
	C 16 MHz RCOSC	C 32.768 kHz XOSC	
PM0	B, C	B or C	ON
PM1	A	B or C	ON
PM2	A	B or C	OFF
PM3	A	A	OFF

*Πίνακας 2*

- PM0: αυτή είναι η κανονική πλήρη λειτουργία του CC2430 με την CPU, όλα τα περιφερικά, τον RF πομποδέκτη, τους ταλαντωτές και εσωτερικούς σταθεροποιητές τροφοδοσίας να είναι όλοι ενεργοποιημένα. Επίσης σε αυτή την κατάσταση λειτουργίας μπορούμε να σταματήσουμε την λειτουργία μόνο της CPU εφόσον δεν χρειάζεται ενεργοποιώντας το bit IDLE του καταχωρητή PCON αλλά όλα τα περιφερικά παραμένουν ως είχαν (δεν απενεργοποιούνται) και η CPU θα αφυπνιστεί από όποιες ενεργοποιημένες διακοπές εμφανιστούν.
- PM1: Στην κατάσταση αυτή απενεργοποιούνται οι γρήγοροι ταλαντωτές και το σύστημα μπορεί να αφυπνιστεί από επανεκκίνηση (HW reset) ή από εξωτερική διακοπή ή τέλος από υπερχείλιση του ειδικού sleep χρονιστή.
- PM2: ίδιο με το PM1 αλλά ακόμα απενεργοποιεί και ο εσωτερικός σταθεροποιητής τροφοδοσίας για ακόμα περισσότερη εξοικονόμηση ενέργειας. Μπορεί να αφυπνιστεί με τους ίδιους τρόπους όπως και με την PM1
- PM3: ίδιο με το PM2 αλλά απενεργοποιεί και τους ταλαντωτές χαμηλής συχνότητας με αποτέλεσμα τα μπορεί να αφυπνιστεί μόνο με εξωτερική διακοπή ή reset.

Σημειώνεται ότι ο ελεγκτής διαχείρισης λειτουργίας δεν απενεργοποιεί τον αναλογικό σταθεροποιητή που τροφοδοτεί ξεχωριστά τον RF πομποδέκτη και αν χρειάζεται να απενεργοποιηθεί πρέπει να γίνει μέσω του καταχωρητή RFPWR.

## WDT

Ο Watchdog timer (WDT) είναι ένας χρονιστής που χρησιμοποιείται κυρίως για τον έλεγχο της σωστής λειτουργίας του λογισμικού που εκτελείται. Το λογισμικό ενδέχεται να “κολλήσει” λόγω παρεμβολών και θορύβου στην τροφοδοσία του CC2430 ή ακόμα και από απροσδιόριστες, μη επιθυμητές καταστάσεις -π.χ. λόγω σφαλμάτων ανάπτυξης- που μπορεί να βρεθεί το λογισμικό. Ο WDT του CC2430 αποτελείται από ένα χρονιστή των 15bits που τροφοδοτείται από 32.768 KHz που δημιουργούνται από τους ταλαντωτές χαμηλής συχνότητας του CC2430. Το λογισμικό έχει σχεδιαστεί κατάλληλα για να αρχικοποιεί την τιμή του WDT σε τακτά χρονικά διαστήματα πριν προλάβει υπερχειλίσει με τρόπο ώστε ο WDT να επανεκκινεί (reset) το CC2430 μόνο σε απροσδιόριστες καταστάσεις “κολλήματος” του λογισμικού. Με τον WDT αυξάνεται η αξιοπιστία εκτέλεσης του προγράμματος γιατί χωρίς την χρήση WDT μπορεί να χρειαζόταν φυσική ανθρώπινη παρέμβαση μετά από ώρες ή και μέρες για την επανεκκίνηση του συστήματος. Στην περίπτωση που δεν απαιτείται ο WDT μπορεί να χρησιμοποιηθεί και ως χρονιστής γενικού σκοπού με δυνατότητα δημιουργίας διακοπής. Γενικά ο WDT χρειάζεται προσοχή γιατί στις καταστάσεις λειτουργίας PM2 και PM3 ο WDT χάνει την 15bit τιμή που είχε και μηδενίζεται.

## USARTs

Το CC2430 εμπεριέχει 2 USART modules USART0 και USART1 με υποστήριξη σύγχρονης και ασύγχρονης σειριακής επικοινωνίας. Τα χαρακτηριστικά των USARTS είναι:

- Επιλογή 8 ή 9 bit αποστολής/λήψης
- Υποστήριξη περιττής ή άρτιας ισοτιμίας
- Ρυθμιζόμενες στάθμες των start stop bits
- Επιλογή αρχικής αποστολής με MSB ή LSB
- Ανεξάρτητη δημιουργία διακοπών σε αποστολή και λήψη
- Ανεξάρτητη πυροδότηση του DMA ελεγκτή σε αποστολή και λήψη
- Έλεγχος σφαλμάτων ισοτιμίας και των σειριακών πλαισίων (framing) με ενημέρωση κατάστασης.
- Υποστήριξη Flow control (RTS,CTS) με χρήση 2 ακόμα pins (4 σύνολο).

Το περιφερειακό USART σε κατάσταση λειτουργίας UART παρέχει full duplex ασύγχρονη σειριακή επικοινωνία δεδομένων με ανεξάρτητο συγχρονισμό των bits του πομπού και δέκτη. Η παραμετροποίηση της λειτουργίας των USARTs γίνεται από τους καταχωρητές UxCSR.

ΑΠΟΣΤΟΛΗ: Η αποστολή των bits από τα USART ξεκινάει όταν φορτωθούν δεδομένα στον διπλό καταχωρητή (double buffered) UxDBUF (το 'x' δηλώνει το χρησιμοποιούμενο USART). Το bit ACTIVE του καταχωρητή UxCSR δηλώνει ότι έχει ξεκινήσει αποστολή bits στο TXDx pin. Όταν ολοκληρωθεί μία αποστολή των 8 ή 9 bits δεδομένων, το bit TX\_BYTE του UxCSR γίνεται λογικό 1. Ο UxDBUF είναι διπλός καταχωρητής και υποστηρίζεται η φόρτωση του επόμενου χαρακτήρα για αποστολή καθώς ο προηγούμενος χαρακτήρας ακόμα αποστέλλεται. Επίσης υποστηρίζει παραγωγή διακοπής κάθε φορά που ο UxDBUF μπορεί να δεχθεί byte για αποστολή.

ΛΗΨΗ: Για να λαμβάνονται σειριακά δεδομένα πρέπει το bit RE του καταχωρητή UxCSR να ενεργοποιηθεί σε λογικό 1. Με το που ανιχνευθεί από το UART ένα έγκυρο bit έναρξης ξεκινάει η λήψη, τα επόμενα bit φορτώνονται στον καταχωρητή λήψης (ολίσθησης) δεδομένων και το bit ACTIVE του καταχωρητή UxCSR μεταβαίνει και παραμένει σε λογικό 1. Μετά από την ολοκλήρωση της λήψης το RX\_BYTE bit θα μεταβεί σε λογικό 1, θα δημιουργηθεί διακοπή και παράλληλα το ACTIVE bit θα μεταβεί σε λογικό 0 αφού ολοκληρώθηκε η λήψη του byte. Το byte αυτό έχει από έχει αποθηκευτεί στον UxDBUF καταχωρητή και μετά από κάθε ανάγνωση του το RX\_BYTE bit μεταβαίνει σε λογικό 1.

ΕΛΕΓΧΟΣ ΡΟΗΣ: Στην περίπτωση που θέλουμε flow control από το UART περιφερειακό πρέπει το bit FLOW του UxCSR καταχωρητή να ενεργοποιηθεί. Έτσι η έξοδος RTS θα οδηγείται σε χαμηλό δυναμικό 0 όταν ο καταχωρητής είναι άδειος και η σειριακή λήψη είναι ενεργοποιημένη. Αντίστοιχα η εκπομπή δεδομένων δεν ξεκινάει μέχρι το CTS σήμα να μεταβεί σε χαμηλό δυναμικό.

ΙΣΟΤΙΜΙΑ: Όταν τα bits BIT9 και PARITY του UxCSR τεθούν σε λογικό 1 τότε ενεργοποιείται η παραγωγή και εκπομπή ενός ακόμα BIT για την ισοτιμία. Αντίστοιχα μετά τη λήψη των bit αυτών παράγεται το bit ισοτιμίας στο δέκτη για να συγκριθεί με το λαμβανόμενο. Εάν υπάρξει λάθος ισοτιμίας τότε το EER bit του UxCSR γίνεται 1. Αυτό το bit χάνεται όμως μετά από ανάγνωση του UxCSR. Ο αριθμός των bit λήξης (stop bits) ρυθμίζεται σε 1 ή 2 από το SPB bit. Ο δέκτης πάντα περιμένει να λάβει 1 bit λήξης και εάν το πρώτο bit λήξης ληφθεί σε λάθος χρονική στιγμή από την προβλεπόμενη τότε το bit FE του UxCSR μεταβαίνει σε λογικό 1 που

δηλώνει σφάλμα συγχρονισμού των bits. Ο δέκτης ρυθμίζεται να ελέγχει 2 λαμβανόμενα bit λήξης όταν το SPB bit έχει ενεργοποιηθεί. Τέλος δημιουργείται διακοπή όταν το πρώτο bit λήξης είναι σωστό και εάν το δεύτερο bit είναι εσφαλμένο τότε θα μεταβαίνει το bit FE του UxCSR σε λογικό 1 αλλά με χρονική καθυστέρηση ενός bit ανάλογα με τον σειριακό ρυθμό μετάδοσης των bits.

Η δημιουργία του ρυθμού μετάδοσης των bits (Baudrate) στην ασύγχρονη σειριακή μετάδοση (UART) γίνεται εφαρμόζοντας την παρακάτω συνάρτηση:

$$Baudrate = \frac{(256 + BAUD\_M) * 2^{BAUD\_E}}{2^{28}} * F$$

Όπου F είναι η συχνότητα λειτουργίας (16MHz RC ή 32MHz XTAL) του CC2430 και BAUD\_M, BAUD\_E είναι 2 καταχωρητές SFR για την επιλογή της επιθυμητής ταχύτητας. Έτσι για παράδειγμα στα 32MHz έχουμε τις παρακάτω γνωστές ταχύτητες με τις παρακάτω τιμές **Πίνακας 3** των καταχωρητών BAUD\_M και BAUD\_E :

Baud rate (bps)	UxBAUD.BAUD_M	UxGCR.BAUD_E	Error (%)
2400	59	6	0.14
4800	59	7	0.14
9600	59	8	0.14
14400	216	8	0.03
19200	59	9	0.14
28800	216	9	0.03
38400	59	10	0.14
57600	216	10	0.03
76800	59	11	0.14
115200	216	11	0.03
230400	216	12	0.03

Πίνακας 3

Παραγωγή διακοπών (interrupts) στο USART: Το CC2430 προσφέρει διαφορετικές και ανεξάρτητες διακοπές σε σχέση με τους κλασική 8051 αρχιτεκτονική και είναι τα URXx και UTXx bits για την λήψη και εκπομπή αντίστοιχα. Στους καταχωρητές IEN0 και IEN2 υπάρχουν τα bits για την ενεργοποίηση παραγωγής διακοπών και στους TCON και IRCON2 τα αντίστοιχα flag bits.

Ποιο συγκεκριμένα τα bits και bit flags για την παραγωγή ή κατά την παραγωγή διακοπών φαίνονται παρακάτω:

**Interrupt enables:**

- USART0 RX : IEN0.URX0IE
- USART1 RX : IEN0.URX1IE
- USART0 TX : IEN2.UTX0IE
- USART1 TX : IEN2.UTX1IE

**Interrupt flags:**

- USART0 RX : TCON.URX0IF
- USART1 RX : TCON.URX1IF
- USART0 TX : IRCON2.UTX0IF
- USART1 TX : IRCON2.UTX1IF

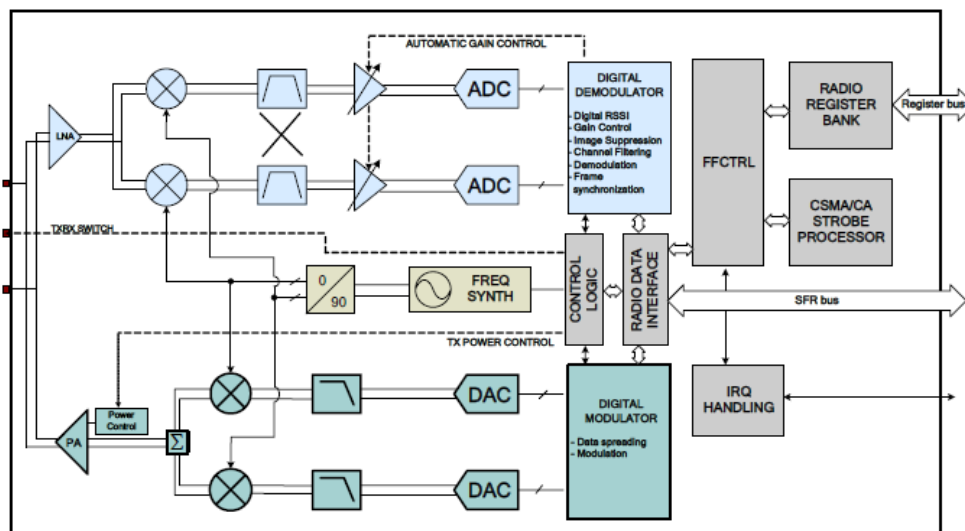
Επίσης στον **Πίνακα 4** περιγράφονται οι διευθύνσεις όλων των διακοπών που υπάρχουν στο CC2430:

Interrupt number	Description	Interrupt name	Interrupt Vector	Interrupt Mask, CPU	Interrupt Flag, CPU
0	RF TX FIFO underflow and RX FIFO overflow.	RFERR	03h	IEN0.RFERRIE	TCON.RFERRIF <sup>7</sup>
1	ADC end of conversion	ADC	0Bh	IEN0.ADCIE	TCON.ADCIF <sup>7</sup>
2	USART0 RX complete	URX0	13h	IEN0.URX0IE	TCON.URX0IF <sup>7</sup>
3	USART1 RX complete	URX1	1Bh	IEN0.URX1IE	TCON.URX1IF <sup>7</sup>
4	AES encryption/decryption complete	ENC	23h	IEN0.ENCIE	S0CON.ENCIF
5	Sleep Timer compare	ST	2Bh	IEN0.STIE	IRCON.STIF
6	Port 2 inputs	P2INT	33h	IEN2.P2IE	IRCON2.P2IF <sup>8</sup>
7	USART0 TX complete	UTX0	3Bh	IEN2.UTX0IE	IRCON2.UTX0IF
8	DMA transfer complete	DMA	43h	IEN1.DMAIE	IRCON.DMAIF
9	Timer 1 (16-bit) capture/compare/overflow	T1	4Bh	IEN1.T1IE	IRCON.T1IF <sup>7,8</sup>
10	Timer 2 (MAC Timer)	T2	53h	IEN1.T2IE	IRCON.T2IF <sup>7,8</sup>
11	Timer 3 (8-bit) compare/overflow	T3	5Bh	IEN1.T3IE	IRCON.T3IF <sup>7,8</sup>
12	Timer 4 (8-bit) compare/overflow	T4	63h	IEN1.T4IE	IRCON.T4IF <sup>7,8</sup>
13	Port 0 inputs	P0INT	6Bh	IEN1.P0IE	IRCON.P0IF <sup>8</sup>
14	USART1 TX complete	UTX1	73h	IEN2.UTX1IE	IRCON2.UTX1IF
15	Port 1 inputs	P1INT	7Bh	IEN2.P1IE	IRCON2.P1IF <sup>8</sup>
16	RF general interrupts	RF	83h	IEN2.RFIE	S1CON.RFIF <sup>8</sup>
17	Watchdog overflow in timer mode	WDT	8Bh	IEN2.WDTIE	IRCON2.WDTIF

*Πίνακας 4*

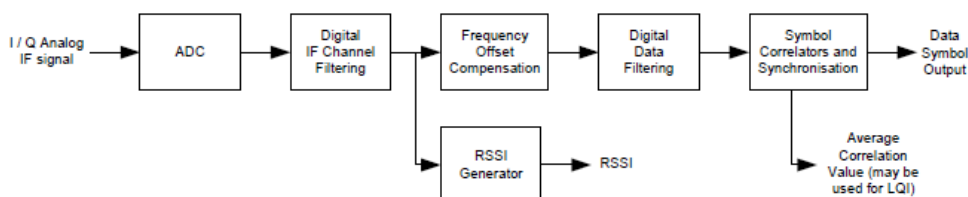
## RADIO

Το ράδιο του CC2430 περιέχει έναν συμβατό πομποδέκτη με το πρότυπο 802.15.4 όπως φαίνεται μπλοκ διάγραμμα, **Εικόνα 3**:



Εικόνα 3

Ο δέκτης λαμβάνει το εισερχόμενο RF σήμα της 2.4GHz ISM μπάντας από την κεραία το ενισχύει με τον ενισχυτή χαμηλού θορύβου (LNA) και υποβιβάζεται (σε I και Q) στην χαμηλή ενδιάμεση συχνότητα (IF). Σε αυτή την σύνθετη χαμηλή συχνότητα των 2MHz, το IF σήμα φιλτράρεται και ενισχύεται για να ψηφιοποιηθεί από τους ADC μετατροπείς του δέκτη. Λειτουργίες όπως αυτόματος έλεγχος κέρδους, τελικό φιλτράρισμα καναλιού, αντιστροφή εξάπλωσης φάσματος (de-spreading), αυτοσυσχέτιση συμβόλων (symbol correlation), προοίμιο και συγχρονισμός σε επίπεδο byte, όλες αυτές είναι υλοποιημένες στο υλικό με ψηφιακό τρόπο. Όταν δημιουργείται διακοπή από τον πομποδέκτη, σημαίνει ότι ο πομποδέκτης έχει αναγνωρίσει την αρχή του πλαισίου (frame delimiter) και έχει αποθηκεύσει το πλαίσιο στην 128byte FIFO (First byte in First byte out) μνήμη λήψης που περιλαμβάνει. Ο χρήστης μπορεί να κάνει ανάγνωση αυτών των δεδομένων μέσω SFR. Είναι προτιμότερο να χρησιμοποιείται ο ελεγκτής DMA για την αντιγραφή των δεδομένων από την RF FIFO μνήμη στην κύρια μνήμη για την μεγιστοποίηση της απόδοσης. Το CRC επαληθεύεται αυτόματα από το HW όπως και η προθήκη στο πλαίσιο των πεδίων RSSI και LQI όπως παρουσιάζεται στο μπλοκ διάγραμμα του αποδιαμορφωτή. **Σχήμα 4**.

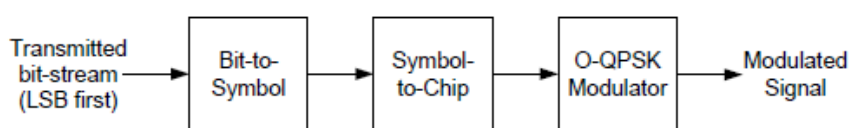


Σχήμα 4

Ο πομπός του CC2430 εκτελεί άμεση ετεροδύωση στην RF συχνότητα εκπομπής. Τα δεδομένα για αποστολή στέλνονται και αποθηκεύονται στην FIFO 128bytes μνήμη αποστολής όπου φυσικά είναι ξεχωριστή και ανεξάρτητη με την FIFO λήψης δεδομένων. Τα πεδία προοιμίου και έναρξης πλαισίου δημιουργούνται σε κάθε πλαίσιο αυτόματα από το υλικό (hardware). Σε κάθε σύμβολο των 4 bits γίνεται ενέργεια εξάπλωσης φάσματος (spread spectrum) σε 32 Chips βάση της ακολουθίας εξάπλωσης όπως ορίζει το πρότυπο IEEE 802.15.4 Πίνακας 5 και μετά οδηγούνται τα Chips ως είσοδο στους DACs μετατροπείς. Ένα αναλογικό χαμηλοδιαβατό φίλτρο διαβιβάζει το σήμα στους ορθογώνιους (I & Q) μίκτες ανοδικής ετεροδύωσης. Το τελικό RF σήμα ενισχύεται από τον ενσωματωμένο ενισχυτή ισχύος και οδηγείται στον διακόπτη επιλογής εκπομπής/λήψης. Το απλοποιημένο μπλοκ διάγραμμα ροής των δεδομένων προς εκπομπή φέρεται στο Σχήμα 5.

Symbol	Chip sequence (C <sub>0</sub> , C <sub>1</sub> , C <sub>2</sub> , ... , C <sub>31</sub> )
0	1 1 0 1 1 0 0 1 1 1 0 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0
1	1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0
2	0 0 1 0 1 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0
3	0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1
4	0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1
5	0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0
6	1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1
7	1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1
8	1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1
9	1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1
10	0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1
11	0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0
12	0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0
13	0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1
14	1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0
15	1 1 0 0 1 0 0 1 0 1 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0

Πίνακας 5



Σχήμα 5

Για την λειτουργία και τον έλεγχο του πομποδέκτη η CPU χρησιμοποιεί ένα σετ από στρόβους εντολών (command strobes), δηλαδή εντολές του ενός byte που ελέγχουν και ρυθμίζουν την λειτουργία του ραδιο-πομποδέκτη. Πρέπει υποχρεωτικά να χρησιμοποιηθούν, ώστε για παράδειγμα να ενεργοποιηθεί ο συνθέτης συχνοτήτων (Frequency Synthesizer), η δυνατότητα εναλλασσόμενης λήψης και εκπομπής από τον πομποδέκτη και άλλες λειτουργίες. Υπάρχουν 9 συνολικά strobes που στέλνονται από την CPU στον CSMA CA/Command Strobe Processor (CSP) προκειμένου να ρυθμίζουν την λειτουργία του πομποδέκτη. Το σέτ εντολών που υποστηρίζει ο CSP παρουσιάζονται στον **Πίνακας 6**

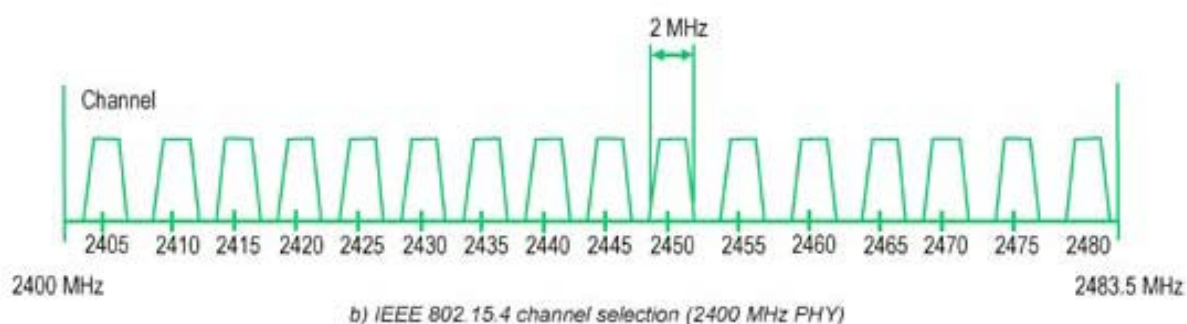
Mnemonic	Opcode Bit number								Description <sup>11</sup>
	7	6	5	4	3	2	1	0	
SKIP C, S	0	S			N	C			Skip S instructions when condition (C xor N) is true. See Table 48 for C conditional codes
WAIT W	1	0	0	W					Wait for W number of MAC Timer overflows. If W is zero, wait for 32 MAC Timer overflows
WEVENT	1	0	1	1	1	0	0	0	Wait until MAC Timer value is greater than or equal to compare value in T2CMP
WAITX	1	0	1	1	1	0	1	1	Wait for CSPX number of backoffs. When CSPX is zero there is no wait.
LABEL	1	0	1	1	1	0	1	0	Label next instruction as loop start
RPT	1	0	1	0	N	C			Repeat from start of loop if condition (C xor N) is true. See Table 48 for C conditional codes
INT	1	0	1	1	1	0	0	1	Assert interrupt
INCY	1	0	1	1	1	1	0	1	Increment CSPY
INCMAXY	1	0	1	1	0	M			Increment CSPY not greater than M
DECY	1	0	1	1	1	1	1	0	Decrement CSPY
DECE	1	0	1	1	1	1	1	1	Decrement CSPZ
RANDXY	1	0	1	1	1	1	0	0	Load CSPX with CSPY bit random value.
Sxxxx	1	1	0	STRB					Command strobe instructions
ISxxxx	1	1	1	STRB					Immediate strobe instructions

Πίνακας 6



Στο φυσικό επίπεδο επειδή τα 250kbit δεδομένων μετατρέπονται σε 2MChips καταλαμβάνεται 2MHz Bandwidth ανά κανάλι, στο υποστηριζόμενο φάσμα εκπομπής λήψης όπως παρουσιάζεται στην **Εικόνα 4**.

Επίσης το πρώτο κανάλι εκπομπής - λήψης από τα 16 σύνολο, έχει ως κεντρική συχνότητα συντονισμού τα 2405 MHz. Τα επόμενα κανάλια είναι τοποθετημένα ανά κάθε 5MHz. Με αυτό τον τρόπο υπάρχει “απόσταση” 3MHz μεταξύ γειτονικών καναλιών.



Source: IEEE 802.15.4 specification

Εικόνα 4

## SIMPLICITI

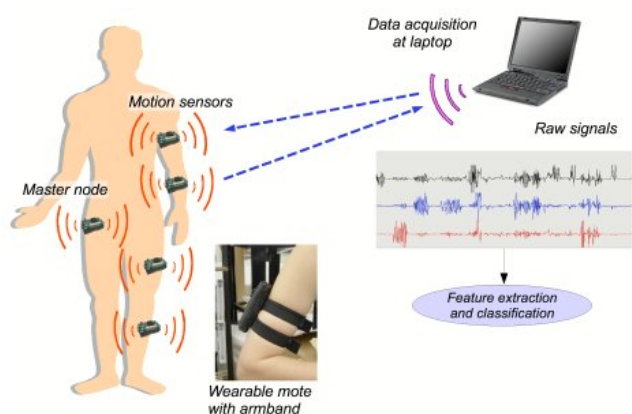
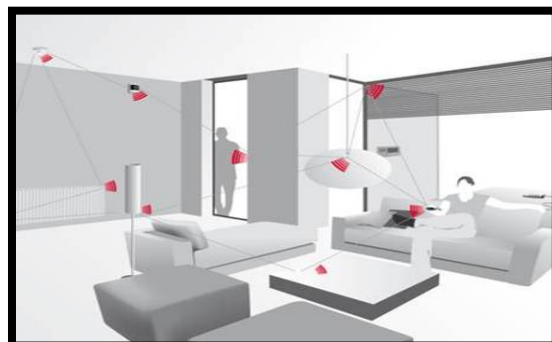
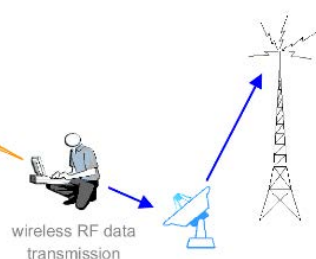
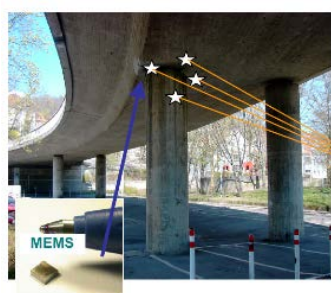
Η SimpliCiTI είναι στοίβα λογισμικού ανοικτού κώδικα ασύρματης επικοινωνίας αισθητήρων, για μικρής κλίμακας ασύρματα δίκτυα αισθητήρων. Είναι σχεδιασμένη και υλοποιημένη αποκλειστικά από την Texas Instruments, με γνώμονα την ευκολία χρήσης και προγραμματισμού, για χαμηλού ρυθμού μετάδοση δεδομένων, χαμηλή κατανάλωση ενέργειας, μικρή γεωγραφική κάλυψη, περιορισμένη υποστήριξη τοπολογιών ασύρματων κόμβων και τέλος για χαμηλό κόστος παραγωγής του τελικού προϊόντος. Διατίθεται στην ιστοσελίδα του κατασκευαστή (TI) για λήψη δωρεάν με διάφορες ενδεικτικές εφαρμογές. Λόγω του ότι παρέχεται σε μορφή ανοικτού κώδικα, προσφέρει στον σχεδιαστή-προγραμματιστή την δυνατότητα πλήρους κατανόησης του τρόπου λειτουργίας της στοίβας, ευκολότερη και ευέλικτη αποσφαλμάτωση σε επίπεδο κώδικα γλώσσας C με εύκολη παραμετροποίηση και μετατροπές ανάλογα με τις ανάγκες της εφαρμογής. Είναι συμβατή με μια μεγάλη γκάμα πομποδεκτών αλλά και SoC ολοκληρωμένων επίσης κατασκευασμένων από την TI όπως το CC2430. Είναι πραγματικά εύκολη στη χρήση μιας και έχει σχεδιαστεί χωρίς την ανάγκη λειτουργικού συστήματος πραγματικού χρόνου (RTOS) και χρησιμοποιεί λίγους πόρους ως προς τις υλικές ανάγκες της.

Μερικά από τα χαρακτηριστικά της είναι:

- Εύκολη στη εφαρμογή της αφού το βασικό API περιέχει μόνο 6 συναρτήσεις που αρκούν για την πλήρη λειτουργία της.
- Μικρές απαιτήσεις σε υλικό (HW) π.χ 16KB Flash και 1KB RAM σε 8051 SoC
- Χαμηλό κόστος και γρήγορη εμφάνιση του προϊόντος στην αγορά
- Χαμηλή κατανάλωση ενέργειας με χρήση διαχείρισης ισχύος CPU & Radio
- Ευρεία συμβατότητα με πολλά ολοκληρωμένα IEEE 802.15.4 συμβατά ή μη.
- Έως και 500Kbps μέγιστου ρυθμού μετάδοσης (RAW data rate)
- Ασφάλεια δικτύου (με συμμετρική κρυπτογράφηση δεδομένων ) όπως XTEA, AES
- Υποστήριξη εύκολης μεταβολής καναλιών για αποφυγή παρεμβολών/θορύβου (Frequency Agility)
- Υποστήριξη εκπομπής/λήψης με μεταπήδηση συχνοτήτων (FHSS)

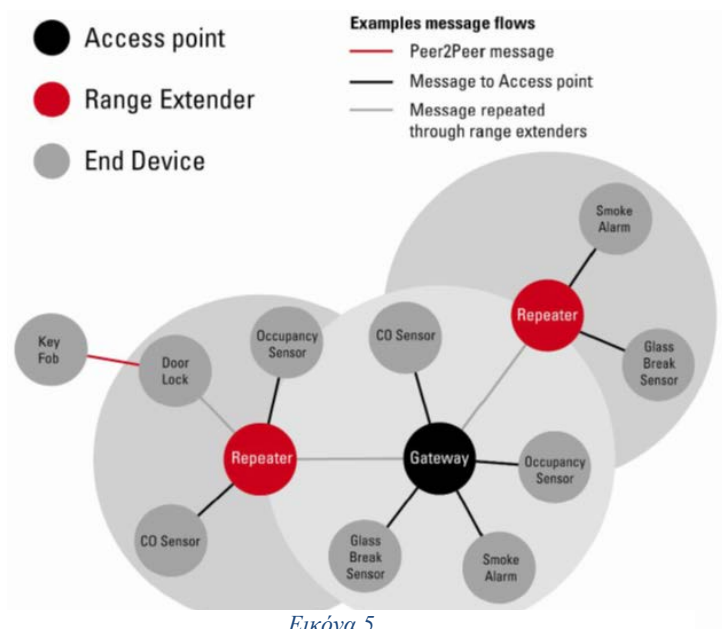
## Παραδείγματα εφαρμογών:

- Σε ασύρματα συστήματα ασφαλείας όπως συναγερμοί, αισθητήρες θραύσης τζαμιών
- Σε διάφορους αισθητήρες όπως CO, Φωτός, Υπέυθρων ή υπερήχων, καπνού κλπ.
- Σε αυτόματους μετρητές όπως υγραερίου, νερού και ηλεκτρικής ενέργειας.
- Σε αυτοματισμούς σπιτιών όπως γκαράζ, αυτόματες πόρτες, παράθυρα κλπ
- Για εφαρμογές ενεργών RFID



## Κατηγορίες κόμβων δικτύου

Σε ένα δίκτυο κόμβων βασισμένο στην SimpliciTI στοίβα υποστηρίζεται η ύπαρξη 3 διαφορετικών κόμβων όπως φαίνεται στην **Εικόνα 5** :



### Τερματική συσκευή ED (End Device)

Η συσκευή ED αποτελεί τον τελικό κόμβο όπου ενσωματώνονται τα αισθητήρια (sensors) και οι ενεργοποιητές (actuators) ανάλογα με τις απαιτήσεις της εκάστοτε εφαρμογής. Η πηγή τροφοδοσίας ενός ED κόμβου είναι συνήθως από μπαταρία/ες ή από ανανεώσιμες πηγές περιορισμένης ενέργειας π.χ. φωτοβολταϊκά πάνελς.

Η ED μπορεί να λειτουργεί συνεχόμενα ή όχι (κατάσταση “ύπνωσης”) για εξοικονόμηση ενέργειας και συνήθως αυτή η δυνατότητα χρησιμοποιείται, όταν η πηγή τροφοδοσίας είναι περιορισμένης ενέργειας.

Οι ED κόμβοι μπορούν να λαμβάνουν και να εκπέμπουν πακέτα ή να εκπέμπουν μόνο χωρίς να λαμβάνουν όπως σε εφαρμογές αποστολής επιλογών χρήστη π.χ. ασύρματοι διακόπτες φωτισμού.

### Σημείο Πρόσβασης (Access Point - AP)

Μόνο ένα AP δύναται να υπάρχει σε ένα δίκτυο εφόσον χρειάζονται οι ειδικές δυνατότητες διαχείρισης του δικτύου και των συσκευών που το απαρτίζουν. Το AP διαχειρίζεται τους κόμβους ED και RE που επιτρέπεται να επικοινωνούν (peer network join) με το διαχειριζόμενο από το AP δίκτυο. Επίσης, διαχειρίζεται τις δυνατότητες υποστήριξης κρυπτογραφημένης επικοινωνίας και Frequency Agility. Επίσης το AP είναι σχεδιασμένο με κύρια πηγή τροφοδοσίας του, το ηλεκτρικό δίκτυο αφού απαιτείται να λειτουργεί ακατάπαυστα για την υποστήριξη ED συσκευών σε κατάσταση ύπνωσης (sleeping mode). Αυτό γίνεται με την αποθήκευση μηνυμάτων δεδομένων από τους κόμβους ED που μεταβαίνουν σε κατάσταση ύπνωσης και την μετέπειτα προώθηση αυτών των μηνυμάτων όταν αιτηθούν από άλλες συσκευές.

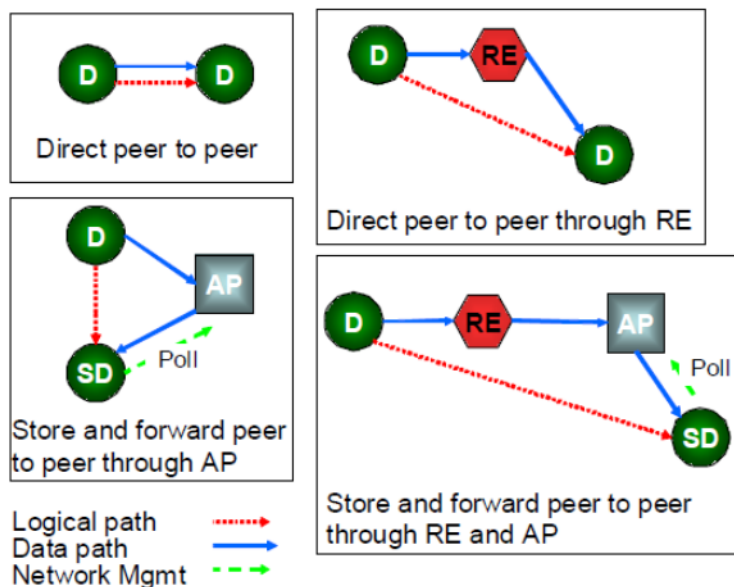
Το AP υποστηρίζει όλες τις δυνατότητες των EDs ως κόμβος αισθητήρων (sensors) και ενεργοποιητών (actuators) όπως και την δυνατότητα να λειτουργεί και ως κόμβος επέκτασης της περιοχής κάλυψης (RE) όπως θα δούμε παρακάτω.

### Range Extender (RE)

Ο κόμβος επέκτασης της κάλυψης δικτύου λειτουργεί ακατάπαυστα αφού είναι και αυτός σχεδιασμένος να έχει ως κυρία πηγή τροφοδοσίας του το ηλεκτρικό δίκτυο. Έχει σχεδιαστεί για την λήψη και αποστολή των λαμβανόμενων πακέτων από τις άλλες συσκευές του δικτύου αλλά αν απαιτείται, μπορεί να ενεργεί και σαν κόμβος ED.

## Τοπολογίες Δικτύου

Οι διάφορες τοπολογίες ενός δικτύου SimpliCI TI που μπορούν να δημιουργηθούν παρουσιάζονται στην **Εικόνα 6** και είναι:



Εικόνα 6

- Άμεση επικοινωνία ED κόμβων: Πρόκειται για την απλούστερη τοπολογία, όπου τα πακέτα δεδομένων ανταλλάσσονται άμεσα μεταξύ των ED κόμβων. Το λογικό μονοπάτι των μεταφερόμενων δεδομένων συμπίπτει με το πραγματικό φυσικό μονοπάτι.
- Άμεση επικοινωνία ED κόμβων διαμέσου RE κόμβου: Το φυσικό μονοπάτι δεδομένων επεκτείνεται λόγω της χρήσης RE κόμβου για την επίτευξη μεγαλύτερης συνολικής κάλυψης δικτύου αφού ο RE κόμβος δρα ως HUB δηλαδή επανεκπέμπει άορατα όλα τα λαμβανόμενα πακέτα ED κόμβων. Το λογικό μονοπάτι είναι ίδιο όπως με την άμεση επικοινωνία ED κόμβων χωρίς RE.
- Αποθήκευση και προώθηση πακέτων ED κόμβων μέσω AP: Ο κόμβος SD είναι απλά ένας ED κόμβος με ενεργοποιημένη την περιοδική ύπωση και όταν είναι ενεργός, στέλνει αίτημα στο AP για να του αποστείλει πακέτα που ενδέχεται να είναι αποθηκευμένα για προώθηση προς αυτό. Το λογικό μονοπάτι είναι το ίδιο όπως στην απλή περίπτωση της άμεσης επικοινωνίας ED κόμβων αλλά το φυσικό μονοπάτι είναι διαμέσου του AP μετά από αιτήματα από το SD στο AP.
- Αποθήκευση και προώθηση πακέτων ED κόμβων μέσω AP και διαμέσου RE: είναι ίδια με την προηγούμενη τοπολογία, με την διαφορά ότι χρησιμοποιούνται RE κόμβοι για την επέκταση της γεωγραφικής κάλυψης του δικτύου.

Από τα παραπάνω αντιλαμβανόμαστε ότι η στοίβα SimpliCI TI είναι σχεδιασμένη να υποστηρίζει μόνο άμεση μεταφορά πακέτων μεταξύ κόμβων με τις παραπάνω παραλλαγές. Αυτό αποτελεί έναν περιοριστικό παράγοντα για ανάπτυξη απαιτητικών εφαρμογών αφού δεν υποστηρίζονται ούτε η δρομολόγηση πακέτων (routing) ούτε οι πλεγματοειδείς τοπολογίες (mesh).

### **Τα επίπεδα της στοίβας.**

Η SimpliCI TI αποτελείται από 3 κύρια επίπεδα: Το επίπεδο εφαρμογής, το επίπεδο δικτύου και το επίπεδο ζεύξης δεδομένων (DataLink - DL) μαζί με το φυσικό επίπεδο.

#### Χαμηλότερο επίπεδο

Έχει αναφερθεί ότι η SimpliCI TI έχει σχεδιαστεί και υλοποιηθεί ώστε να είναι συμβατή για να εκτελεστεί από μία ευρεία γκάμα μικροελεγκτών με ενσωματωμένο πομποδέκτη. Λόγω αυτής της δυνατότητας, η οργάνωση του χαμηλότερου επιπέδου της στοίβας είναι χωρισμένη σε δύο κατηγορίες. Με αυτόν το τρόπο δίνεται η δυνατότητα στον σχεδιαστή – προγραμματιστή να χρησιμοποιεί την στοίβα με ελάχιστες έως και μηδενικές τροποποιήσεις στον κώδικα επιπέδου εφαρμογής.

Οι δύο κατηγορίες οργάνωσης του χαμηλότερου επιπέδου είναι:

Board Support Package (BSP): Είναι η κατηγορία αρχείων C κώδικα της στοίβας (αντίστοιχη με την κατηγορία Hardware Abstraction Layer), υπεύθυνη για την βασική υποστήριξη, αρχικοποίηση και παραμετροποίηση στοιχείων για την σωστή λειτουργία μεταξύ διάφορων αρχιτεκτονικών μικροελεγκτή όπως ο 8bit 8051 και ο 16bit MSP430. Πιο συγκεκριμένα η συνάρτηση BSP\_init() χρησιμοποιείται για την αρχικοποίηση (initialization) όλων των απαιτούμενων στοιχείων όπως: CPU endianness, διαχείριση ταλαντωτών, ρύθμιση και συγχρονισμός χρονιστών για την παραγωγή συγκεκριμένων χρονοκαθυστερήσεων, διαχείριση διακοπών, ενεργοποίηση και αρχικοποίηση διαφόρων περιφερειακών διατάξεων όπως ακροδέκτες για είσοδο/έξοδο δεδομένων και σειριακή επικοινωνία (I2C, SPI, UART). Ιδιαίτερα, η σειριακή σύγχρονη επικοινωνία SPI χρησιμοποιείται για την επικοινωνία μεταξύ MCU και πομποδέκτη στις περιπτώσεις που χρησιμοποιούνται ως διαφορετικά φυσικά ολοκληρωμένα και όχι στην υλοποίηση SoC.

Minimal RF interface (MRFI): Είναι η αναγκαία κατηγορία αρχείων κώδικα που υλοποιεί την βασική επικοινωνία φυσικού επιπέδου για όλους τους συμβατούς πομποδέκτες με τις ανάλογες αρχικοποιήσεις των ειδικών καταχωρητών των πομποδεκτών. Συγκεκριμένα οι συναρτήσεις MRFI δέχονται τα δεδομένα από ανώτερα επίπεδα (επίπεδο δικτύου) ενθυλακώνοντάς τα με τις κατάλληλες κεφαλίδες. Έτσι, δημιουργούνται τα MRFI πλαίσια δεδομένων όπου οι αντίστοιχες συναρτήσεις χρησιμοποιούν τον πομποδέκτη για αποστολή και λήψη τους. Επίσης το MRFI είναι υπεύθυνο για την προσπέλαση μέσου (MAC) με την μέθοδο Clear Channel Assessment (CCA), ελέγχοντας δηλαδή την διαθεσιμότητα του καναλιού για την ελαχιστοποίηση των συγκρούσεων και συνάμα την μεγιστοποίηση της μετάδοσης των δεδομένων.

Επίπεδο δικτύου (NWK):

Το επίπεδο NWK ενθυλακώνει τα δεδομένα του επιπέδου εφαρμογής και τοποθετεί NWK κεφαλίδα για υποστήριξη διαφορετικών συνδέσεων. Επίσης το επίπεδο NWK διαχειρίζεται και ελέγχει μια μνήμη, τύπου ουράς για τα λαμβανόμενα και εκπεμπόμενα πλαίσια NWK (frame queue). Τέλος διαχειρίζεται και χρησιμοποιεί το MRFI επίπεδο για την προώθηση και παραλαβή NWK πακέτων. Στο πλαίσιο του επιπέδου NWK χρησιμοποιείται το πεδίο port number που καταλαμβάνει 1 byte από την NWK κεφαλίδα και χρησιμοποιείται για τον καθορισμό της εφαρμογής. Υπάρχουν μερικές εφαρμογές με συγκεκριμένα port numbers για αποκλειστική χρήση από το επίπεδο NWK και αυτές είναι:

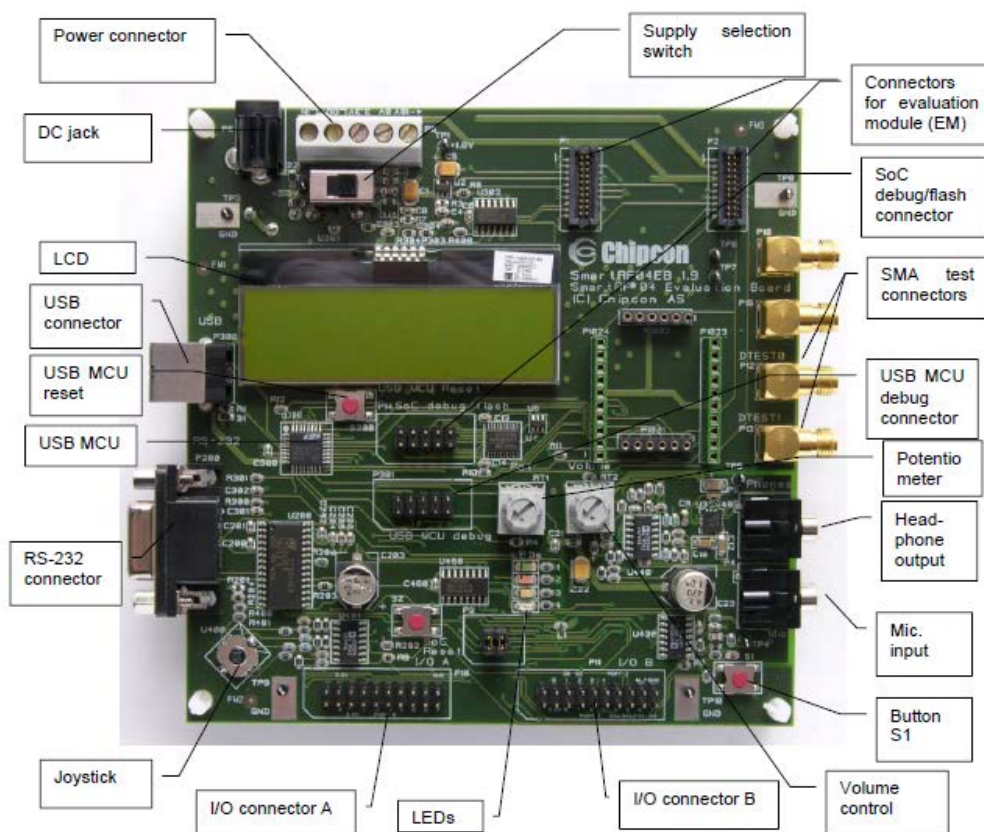
- Ping με πόρτα (0x01): Χρησιμοποιείται ως η μοναδική εφαρμογή για την ανίχνευση της παρουσίας μιας συσκευής σε ένα δίκτυο.
- Link με πόρτα (0x02): Υποστηρίζει την διαχείριση σύνδεσης μεταξύ 2 συσκευών
- Join με πόρτα (0x03): Χρησιμοποιείται για την φύλαξη και προστασία εισόδου σε ένα δίκτυο με AP
- Security με πόρτα (0x04): Χρησιμοποιείται για την ανταλλαγή πληροφοριών σχετικά με την ασφάλεια του δικτύου όπως στην αλλαγή των κωδικών (κλειδιών) κρυπτογράφησης και των κρυπτογραφημένων δεδομένων.



## ΠΕΙΡΑΜΑΤΙΚΗ ΕΦΑΡΜΟΓΗ

Έχουν περάσει 4 δεκαετίες όταν πρωτοεμφανίστηκε το πρότυπο σειριακής επικοινωνίας RS232 αλλά αρκετές σχεδιάσεις συνεχίζουν να το ενσωματώνουν ακόμα και μετά την έλευση της USB σειριακής τεχνολογίας. Παράλληλα, με την όλο και αυξανόμενη διάδοση των ασύρματων δικτύων αισθητήρων αλλά κυρίως για τον λόγο ότι έχει μειωθεί το κόστος κατασκευής τους, ιδιαίτερα στις SoC πλατφόρμες, προέκυψε το ερώτημα αν είναι εφικτή η χρήση της ασύρματης τεχνολογίας για την μεταφορά RS232 δεδομένων. Πιο συγκεκριμένα η χρήση ενός ασύρματου δικτύου αισθητήρων για την μεταφορά σειριακών δεδομένων. Όμως οι ασύρματες επικοινωνίες μεταφοράς δεδομένων είναι ποιο επιρρεπείς σε θόρυβο και παρεμβολές γιατί το μέσο μετάδοσης που χρησιμοποιείται σε αυτές είναι ο αέρας στην περίπτωση μας.

Για την εφαρμογή χρησιμοποιήθηκαν 2 αναπτυξιακά SmartRF04EB μαζί με 2 CC2430EM boards. Στην **Εικόνα 7** παρουσιάζεται το αναπτυξιακό και περιγράφονται μερικές από τις λειτουργίες του.

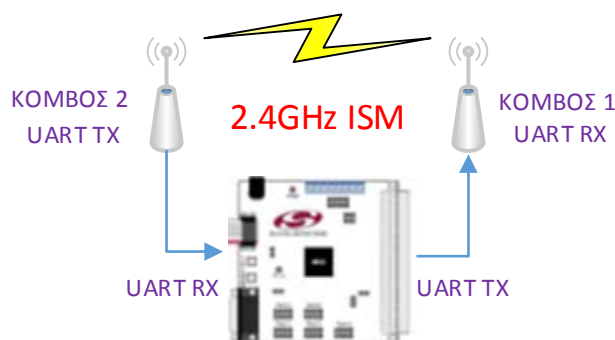


Εικόνα 7

Για την ανάπτυξη της πειραματικής εφαρμογής χρησιμοποιήθηκε η SimpliCITY έκδοση 1.2 για 8051 και το πρόγραμμα ανάπτυξης σε και αποσφαλμάτωσης σε γλώσσα C IAR

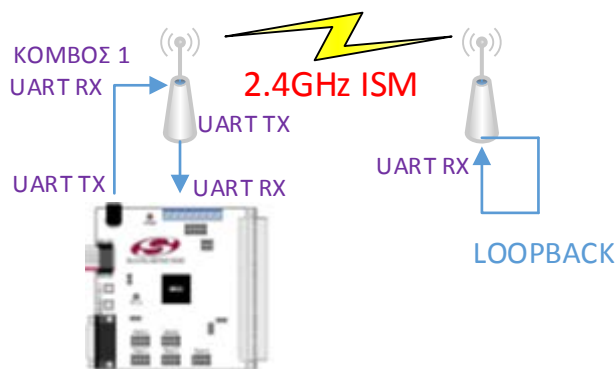
embedded workbench 8051 έκδοσης 8.20.2 στην δωρεάν δοκιμαστική έκδοση των 30ημερών. Επίσης χρησιμοποιήθηκε και το C8051F020-DK αναπτυξιακό από την Silicon Labs με τον C8051F020 μικροελεγκτή για τη παραγωγή συγκεκριμένων σειριακών δεδομένων κατά βούληση μέσω διακόπτη και των έλεγχου της λήψης τους με τους παρακάτω τρόπους συνδεσμολογίας:

Πρώτος τρόπος συνδεσμολογίας (A1): η έξοδος TX του C8051F020 συνδέθηκε στην είσοδο RX του πρώτου κόμβου (SmartRF04EB αναπτυξιακό) για την αποστολή των δεδομένων και η RX είσοδος του C8051F020 συνδέθηκε στην έξοδο TX του δεύτερου κόμβου για τον έλεγχο των λαμβανόμενων δεδομένων μέσω ασύρματης επικοινωνίας. Με αυτή τη συνδεσμολογία ελέγχεται η αποστολή δεδομένων προς μια κατεύθυνση από τον κόμβο 1 προς στον 2 μόνο. Η συνδεσμολογία περιγράφεται στην **Εικόνα 8**



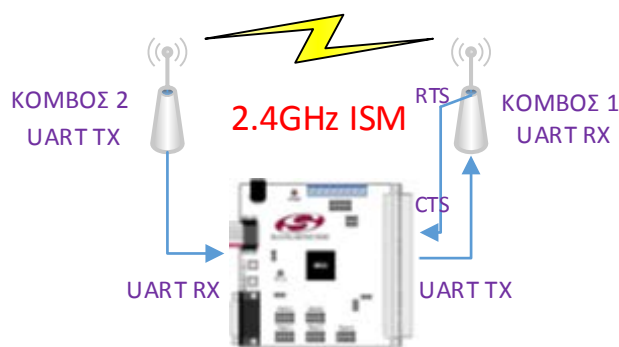
Εικόνα 8

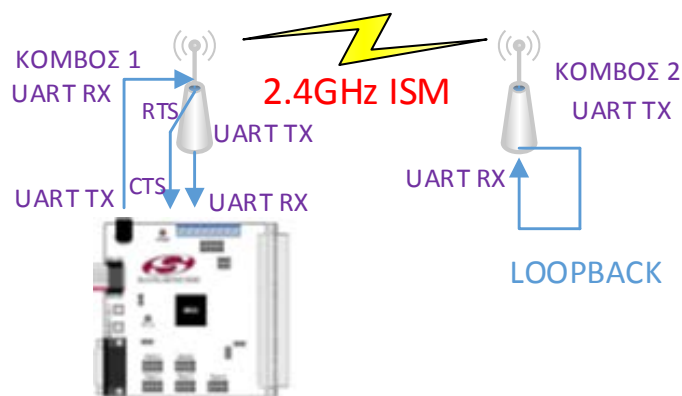
Δεύτερος τρόπο συνδεσμολογίας B1: οι ακίδες TX και RX του CC8051F020 συνδέθηκαν εναλλάξ με το πρώτο κόμβο (SmartRF04EB αναπτυξιακό) και στο δεύτερο κόμβο οι ακίδες TX και RX βραχυκυκλώθηκαν. Με αυτόν το τρόπο όλα τα λαμβανόμενα δεδομένα στον δεύτερο κόμβο που στέλνονται από τον πρώτο κόμβο θα σταλούν στην TX έξοδο του αλλά ουσιαστικά θα ληφθούν αμέσως από την RX είσοδο (loopback) ώστε ο δεύτερος κόμβος να τα στείλει πίσω στο πρώτο κόμβο. Με αυτή τη συνδεσμολογία **Εικόνα 9** ελέγχεται η αποστολή δεδομένων δύο κατευθύνσεων (full duplex) από τον κόμβο 1 στον 2 και από 2 σε 1 ταυτόχρονα μετά από την πρώτη λήψη δεδομένων από τον κόμβο 2.



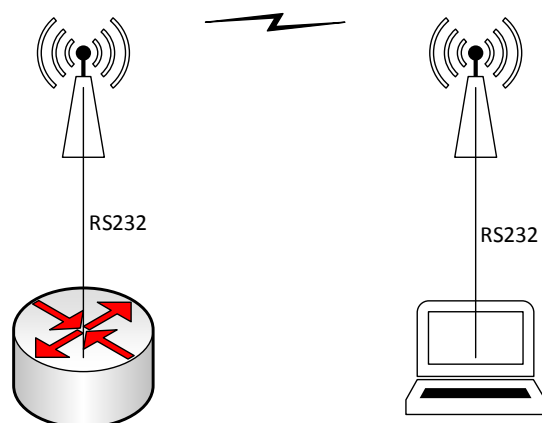
Εικόνα 9

Επίσης έγιναν ξανά οι προηγούμενες πειραματικές δοκιμές αλλά με έλεγχο ροής ώστε να σταματάει η αποστολή των σειριακών RS232 δεδομένων από το C8051F02x-DK ώστε να προστατεύεται η επικοινωνία από υπερχειλίση δεδομένων των UART Buffers των ασύρματων κόμβων. Αυτό επιτυγχάνεται όταν το Buffer του κόμβου είναι σχεδόν γεμάτο το RTS μεταβαίνει σε επίπεδο MARK. Με αυτόν τον τρόπο σταματάει προσωρινά η αποστολή-λήψη RS232 δεδομένων μέχρι να αποσταλούν τα δεδομένα αυτά ασύρματα. Εφόσον η στάθμη του UART Buffer του κόμβου ελαττωθεί αρκετά και γίνει μικρότερη από μία τιμή “threshold” τότε ο κόμβος μεταβαίνει το RTS σε επίπεδο SPACE για να συνεχίσει η αποστολή-λήψη RS232 δεδομένων. Οι συνδεσμολογίες A2, B2 είναι παρόμοιες με τις προηγούμενες A1, B1 αλλά με έλεγχο ροής ενεργοποιημένο (μονής κατεύθυνσης) και παρουσιάζονται στις εικόνες 10 και 11 αντίστοιχα:





Ως τελικό στάδιο της πειραματικής εφαρμογής χρησιμοποιήθηκε ένας Cisco Router σειρά 2600 ώστε να αξιολογηθεί η ποιότητα μεταφοράς των RS232 δεδομένων βασισμένη σε πραγματική εφαρμογή **Σχήμα 6**



Σχήμα 6

Σημειώνεται ότι όλες η δοκιμές έγιναν με ταχύτητα σειριακής μετάδοσης δεδομένων (baud rate) 9600bps.

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Οι ασύρματες ζεύξεις λειτουργούν ικανοποιητικά σε πραγματικές συνθήκες αλλά υπάρχουν προβλήματα χαμένων/αλλοιωμένων δεδομένων ακόμα και με μηχανισμούς επιβεβαίωσης ACKs. Το πρόβλημα είναι ιδιαίτερα εμφανές στην συνδεσμολογία πλήρως αμφίδρομης σειριακής επικοινωνίας χωρίς έλεγχο ροής. Αυτό συμβαίνει γιατί τα λαμβανόμενα σειριακά δεδομένα που αποθηκεύονται στην μνήμη Xdata (Buffering) περιορισμένης χωρητικότητας έως 2,5Kbytes, υπερχειλίζουν σε συνεχόμενη αποστολή-λήψη RS232 δεδομένων λόγω μη επαρκούς απόδοσης της ασύρματης ζεύξης. Είναι προφανές, ότι η ασύρματη ζεύξη μεταφέρει δεδομένα προς μια κατεύθυνση κάθε φορά δηλαδή είναι Half duplex και συνεπάγεται με καθυστέρηση μετάβασης του ενσωματωμένου πομποδέκτη στις λειτουργίες εκπομπής-λήψης άρα και περιορισμό της ταχύτητας επικοινωνίας. Στα γραφήματα του παράρτηματος (**σελίδα 63**), παρουσιάζονται οι μέγιστες αποδόσεις τριών διαφορετικών τεχνολογιών ασύρματων επικοινωνιών υπο ιδανικές συνθήκες επικοινωνίας (Υψηλό SNR & Signal Interference Ratio).

Ένας βασικός περιορισμός κατά τη διάρκεια των πειραματικών δοκιμών ήταν η μη διάθεση ενός τρίτου ενδιάμεσου κόμβου και την λειτουργία αυτού ως μόνιμο δέκτη για την λήψη όλων των πακέτων μεταξύ των κόμβων. Με αυτόν το τρόπο θα μπορούσαν να εκτιμηθούν ως ένα βαθμό, τα προβλήματα απόδοσης της ασύρματης ζεύξης αφού θα μπορούσαν να ανιχνευθούν λανθασμένα πακέτα και επανεκπομπές πακέτων ανάλογα του περιβάλλοντος θορύβου – παρεμβολών στις πειραματικές εφαρμογές.

Μερικοί τρόποι βελτίωσης της ασύρματης ζεύξης είναι:

- Ο συγχρονισμός των 2 κόμβων ασύρματης επικοινωνίας τύπου “beacon” όπου οι δύο κόμβοι αυτοσυγχρονίζονται στο πότε θα εκπέμπουν και θα λαμβάνουν εναλλάξ για να εξαλειφθούν οι περιπτώσεις όπου και οι δυο κόμβοι προσπαθούν να στείλουν ασύρματα δεδομένα την ίδια χρονική στιγμή.
- Η χρησιμοποίηση αλγόριθμου (Handshaking) επιλογής προτεραιότητας κατεύθυνσης μεταφερόμενων δεδομένων ανάλογα με τις στάθμες buffer των κόμβων.
- Η χρήση τεχνικής FHSS για αύξηση της ανοχής σε θόρυβο – παρεμβολές.

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

CC2430 Datasheet, SWRS036F Texas Instruments

Designer's Guide to LPRF, SLYA020A Texas Instruments

Introduction to SimpliciTI SWRU130b, Texas Instruments

SimpliciTI Overview SWRU130, Texas Instruments

SimpliciTI API Texas Instruments

SimpliciTI Specification Texas Instruments

SimpliciTI Developers Notes Texas Instruments

CC2430 Software Examples User's Guide SWRU178B, Texas Instruments

Power Modes in CC111xFx, CC243x, and CC251xFx DN106 Texas Instruments

<http://www.dragonwins.com/domains/getteched/de248/rs232.htm>

IEEE, *Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)*. New York: Institute of Electrical and Electronics Engineers, 2003.

**K. Mikhaylov, N. Plevritakis, J. Tervonen**, "*Performance Analysis of Bluetooth Low Energy in Wireless Sensor Networks Applications*" (DRAFT)

ΠΑΡΑΡΤΗΜΑ

# Performance Analysis of Bluetooth Low Energy in Wireless Sensor Networks Applications (DRAFT)

Authors:

Konstantin Mikhaylov, Oulu Southern Institute, University of Oulu

Email: [konstantin.mikhaylov@oulu.fi](mailto:konstantin.mikhaylov@oulu.fi)

Nikolaos Plevritakis, Department of Electronic Engineering, Technological Educational Institute of Crete

Email: [nplevis@chania.teicrete.gr](mailto:nplevis@chania.teicrete.gr) , [nplevr@gmail.com](mailto:nplevr@gmail.com)

Jouni Tervonen, Oulu Southern Institute, University of Oulu

Email: [jouni.tervonen@oulu.fi](mailto:jouni.tervonen@oulu.fi)

## Abstract

**The Bluetooth Low Energy (BLE) is one of the most recently developed protocols for short range wireless communication that is predicted to find wide application in Wireless Sensor Networks (WSNs). In the paper, we present the results of the empirical measurements for such parameters as the required resources, the maximum peer-to-peer throughput, the minimum round trip time and the energy per packet and per byte consumption for the real-life hardware and software stacks implementing three wireless communication protocols, namely the BLE (TI BLE protocols stack), the IEEE 802.15.4 (TIMAC stack) and the company-proprietary protocol (SimpliciTI stack) from Texas Instruments. Based on the presented measurement and made analysis' results, we compare BLE with the existing protocols and draw the conclusions about the feasibility and the most suitable application scenarios of the current BLE implementations in WSN applications.**

**Keywords:** *Wireless Sensor Networks; WSN; WSN; Bluetooth Low Energy; BLE; performance; energy; efficiency; measurements; consumption; 802.15.4; ZigBee; TIMAC; SimpliciTI*

## **Introduction**

During the recent years, the Wireless Sensor Networks (WSNs) has attracted very serious attention both from the academy and from the industry and have been named among the key technologies for the future. Today, the WSNs are becoming the integral part of everyday life and help to solve different tasks for the broadest range of applications operating at various environment conditions. Nonetheless, the diversity of the WSN applications and the difference in their requirements prevent the unification and standardization for the technologies and solutions used in the WSNs.

This is one of the major reasons why nowadays the broad range of the wireless communication protocols for WSNs exist and why some more protocols are currently being developed. Bluetooth Low-Energy (BLE) is one of the most recently developed protocols, which is aimed also for the

WSN applications. It has been introduced as a part of IEEE 802.15.1 Bluetooth Core Specification 4.0 in 2010 [1]. It has been predicted [2] that by the year 2015 BLE will dominate the WSN market. Nonetheless, even today, i.e., two years after the finalization of the BLE specification and almost one year after the appearance of the first commercial radio chips implementing the BLE, it is still not really shown clearly whether the BLE protocol itself and the current hardware (HW) and software (SW) implementations of the BLE are well feasible for WSN applications. Therefore, in this paper, we present the results of the performance measurements for the BLE and compare those with the results of IEEE 802.15.4 and a proprietary protocol based on Frequency Shift Keying (FSK). The specifics and the main contribution of this research is that it composes the heuristic analysis of the available protocols and of their features and the results of empirical measurements for the real HW devices implementing the protocols discussed. Based on the measurement results, we evaluate the suitability of BLE for the WSNs and define the most suitable application areas. The paper is organized as follows. **Section II** discusses the related work. **Section III** provides the short overview of the most widely used WSN communication protocols and reveals the key details of the tested protocols. **Section IV** describes the measurement procedure and presents the obtained measurement results. Finally, **Section V** summarizes the obtained results and provides the analysis of the BLE application possibility for the WSN applications.



Table I: Features of the most widely used communication protocols for the WSN applications <sup>1</sup>

Protocol name	Year <sup>1</sup>	Purpose	Standardized layers <sup>4</sup>	Commercial devices	Frequency bands	Modulation	Channel spreading method	Symbol rate, kbit/s	Max. packet payload	Channel access method	Network topology	Multi-hopping support	Number of nodes
IEEE 802.15.4	2003	low-rate wireless personal area networks	PHY, DL(MAC)	available	868/915 MHz, 2.4 GHz	BPSK/O-QPSK	DSSS	<250	127 bytes	CSMA/CA	P2P, star, mesh, hybrid	supported	n/a
ISO 18000-7 (DASH7)	2004	ultra low energy wireless communication	PHY, DL	unavailable <sup>5</sup>	433 MHz	FSK/GFSK	Not applicable	27.78 <sup>8</sup>	249 <sup>8</sup> bytes	slotted aloha	master-slave	limited support	n/a
IEEE 802.15.1 (Bluetooth)	1998	wireless personal area network	All layers	available	2.4 GHz	GFSK	FHSS	1000	2745 bits	FH-CDMA/TDMA	master-slave	limited support	7 active per piconet
IEEE 802.11b/g	1999	wireless local-area-networks	PHY, DL (LLC + MAC)	available	2.4 GHz	DBPSK/DQPSK	DSSS	11000/54000	2304 bytes	CSMA/CA	P2P, star, mesh, hybrid	supported	unlimited
IEEE 802.15.4a	2007	wireless personal area networks with ranging capability	PHY, DL(MAC)	under development	250-750 MHz, 3-5 GHz, 6-10 GHz	BPSK-BPPM	TH-IR UWB	27240	1209 symbols	CSMA/CA or slotted aloha	P2P/star	supported	<65536
IEEE 802.15.1-4 (Bluetooth Low Energy)	2010	low-power and low-latency short range wireless communication	All layers	available	2.4 GHz	GFSK	FHSS	1000	23 bytes	FH-CDMA/TDMA	star	not supported	unlimited
ISO/IEC 14543-3-10 (EnOcean)	2012	wireless solutions with ultra-low power consumption and energy harvesting for building automation	PHY, DL, NWK	available	315/868 MHz	ASK	n/a <sup>2</sup>	125	not defined	CSMA/CA	P2P, star, mesh, hybrid	supported	>65536
ITU G.9959 (Z-Wave)	2012	wireless communications protocol for home automation	PHY, DL(MAC)	available	868/915 MHz	FSK/GFSK	n/a <sup>2</sup>	100	158 bytes	CSMA/CA	P2P, star, mesh, hybrid	source routing	232 per network
IEEE 802.15.6	2012	wireless body area networks	PHY, DL(MAC)	unavailable	400/433 /868/915 MHz, 2.4 GHz	numerous <sup>6</sup>	Several options	<15600	255 bytes	CSMA/CA /slotted aloha/TDMA	star	limited support (up to 2 hops)	64
Proprietary protocols	n/a <sup>2</sup>	n/a <sup>2</sup>	PHY, sometimes DL(MAC) or NWK	available	400/433 /868/915 MHz, 2.4 GHz	mostly often: ASK, FSK, GFSK	Several options	usually <2000	usually <255	CSMA/CA /slotted aloha/TDMA	n/a <sup>2</sup>	n/a <sup>2</sup>	n/a <sup>2</sup>
IEEE 802.11ah	2015 <sup>3</sup>	sub 1 GHz sensor and smart metering network	PHY, DL(MAC)	-	868/915 MHz	Tbd <sup>7</sup>	Tbd <sup>7</sup>	Tbd <sup>7</sup>	Tbd <sup>7</sup>	Tbd <sup>7</sup>	Tbd <sup>7</sup>	Tbd <sup>7</sup>	Tbd <sup>7</sup>

<sup>1</sup> – year of initial specification release<sup>2</sup> – Not applicable<sup>3</sup> – currently is under development, release is planned for 2015<sup>4</sup> – PHY – physical, DL – data-link, MAC – media access control, LLC – logical link control, NWK – network<sup>5</sup> – commercial radios compatible with DASH7 PHY are existing, but those do not include the appropriate DASH7 stack implementation<sup>6</sup> – depending on the mode can be used: DBPSK, DQPSK, D8PSK, GMSK, OOK, CP-BFSK, DPSK, FM-UWB<sup>7</sup> – TBD<sup>8</sup> – DASH7 Mode 2 (2009) supports the data rate of 250 kbit/s and packet payload of 249 bytes<sup>1</sup> The table summarizes the data obtained from [3–21]

## Related work

Although quite significant time had passed since the finalization of BLE specification, still there are not so many published research works considering BLE. The existing works considering the BLE can be divided into four major groups.

The first group includes the papers that suggest the novel methods for implementing some hardware components of the BLE transceiver (e.g., the demodulator [22], [23]) or the transceiver as a whole [24].

The papers in the second group focus on the process of design and provide valuable data concerning the implementation of the end-devices that use the BLE for communication. The majority of the currently available papers (e.g., [14], [25], [26]) introduce the BLE-based devices that target various healthcare applications.

The authors of the papers in the third group use mathematical analysis and simulations to model the BLE operation and estimate the required performance parameters. E.g., in [15] the authors present the analytical model for BLE maximum throughput and investigate the influence of errors and connection interval length on the maximum throughput. In [27] the authors introduce the analytical model for 3-channel-based neighbor discovery for BLE and use it for defining the performance metrics such as the average latency or the average energy consumption during the procedure of neighbors discovering.

Finally, the last group of publication constitutes of the presentations of the empirical measurements results for the real-life BLE HW and SW. Some of the data is provided by the BLE radio manufacturers as application or design notes or white papers (e.g., [28]), while [29] represent the only scientific research paper that suits this group we able to find. However, the major focus for these papers is typically specifically the power consumption of BLE (see [28], [29]) while all the other performance characteristics that are vital e.g., for estimating the suitability of BLE for particular application, are usually left aside.

### Communication protocols in WSNs.

Due to the reasons discussed in [Section I](#), nowadays there are multiple communication protocols that are used by the WSNs. To help understanding the specifics of BLE compared with the other existing and incoming WSN protocols, [Table I](#) provides the short summary of the protocol features. Due to the limited space, we will not discuss all these protocols in detail; instead we will focus on the three protocols that were used during measurements, namely the IEEE 802.15.4, the proprietary FSK-based protocols, and the BLE.

#### IEEE 802.15.4

The IEEE 802.15.4 [30] is the most widely used communication protocol and can be considered as de-facto the standard for the WSNs [31], [32]. The IEEE 802.15.4 standard (we consider IEEE 802.15.4-2006, unless noted especially) defines the physical (PHY) and part (namely media access control (MAC)) of the data-link (DL) layers for the widest range of wireless personal area networks (WPANs) applications with mild throughput and latency requirements. The main features of the standard are the low implementation complexity, low costs and low power consumption [32], which caused the wide utilization of IEEE 802.15.4 especially for the networks with low traffic and which include plenty of cheap nodes. In the recent years, several upper layers protocols, such as ZigBee, RF4CE, 6LowPAN, MiWi, WirelessHART and many vendor-specific ones, have been developed based and above IEEE 802.15.4.

The PHY layer of IEEE 802.15.4 operates in three bands within unlicensed industrial, scientific and medical (ISM) radio bands, namely 868.0-868.6 MHz (Europe), 902-928 MHz (North America) and 2400-2483.5 MHz (worldwide) [30]. Based on the used frequency band, the protocol dictates to use different modulation (BPSK, O-QPSK or ASK for 868/915 MHz bands and O-QPSK for 2.4 GHz) and different spectrum spreading techniques (DSSS or PSSS for 868/915 MHz

bands, DSSS – for 2.4 GHz). The structure of IEEE 802.15.4 PHY frame is revealed in Fig. 1a. The frame consists of the synchronization header (SHR) for providing the bit and byte-wise synchronization, PHY header (PHR) revealing the size of the whole packet and, if required, the Protocol Data Unit (PDU).

The MAC sub-layer of IEEE 802.15.4 defines the access to the media, the format for different frames and some other features (e.g., addressing and frame validation). The standard [33] includes four possible types of frames with appropriate MAC formats: the beacon frame, the data frame, the acknowledgement (ACK) frame and the MAC command frame. The MAC frames (see Fig. 1a) consist of the MAC header (MHR) containing the information about the frame type and other data regarding frame processing, the unique frame identifier (ID) and the data about frame source and destination, the MAC data unit (MSDU) and the MAC footer with the check-sequence. The IEEE 802.15.4 defines two different operation modes with different channel access mechanisms: the beacon enabled and non-beacon enabled. The former one assumes that the WPAN includes the coordinator device that periodically transmits the beacon-packets that are used to define the boundaries of the superframes. Generally, a superframe can include two major phases: the contention-free period (CFP) where the time slots within CFP are booked for the transmission of specific nodes and the contention-access period (CAP) during which the nodes compete (using Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)) to transmit the packet. After the superframe the coordinator can define the inactive period, when no communication is possible and the nodes can switch off the radio to save energy. In the non-beacon enabled mode the nodes use the CSMA/CA to access the channel and transmit their packets whenever required [32].

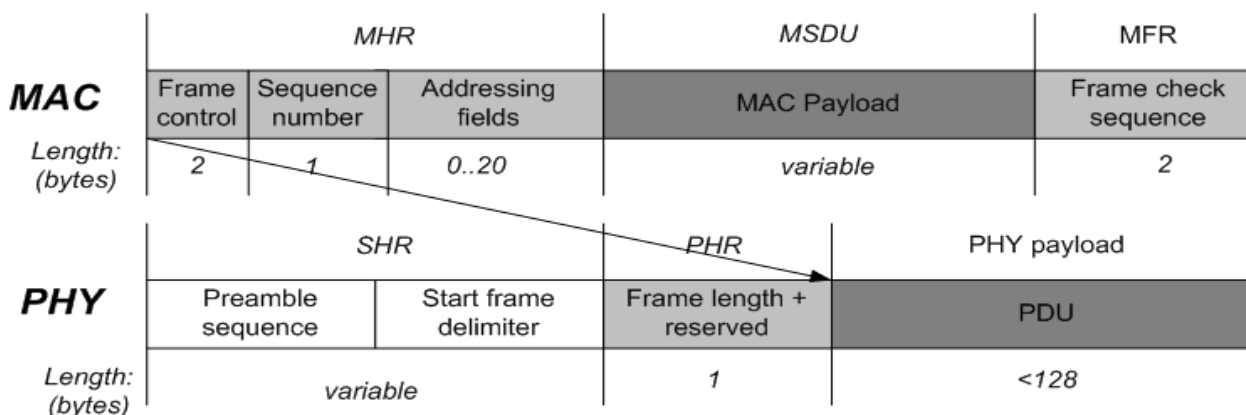


Fig. 1 a. Data frame format for IEEE 802.15.4

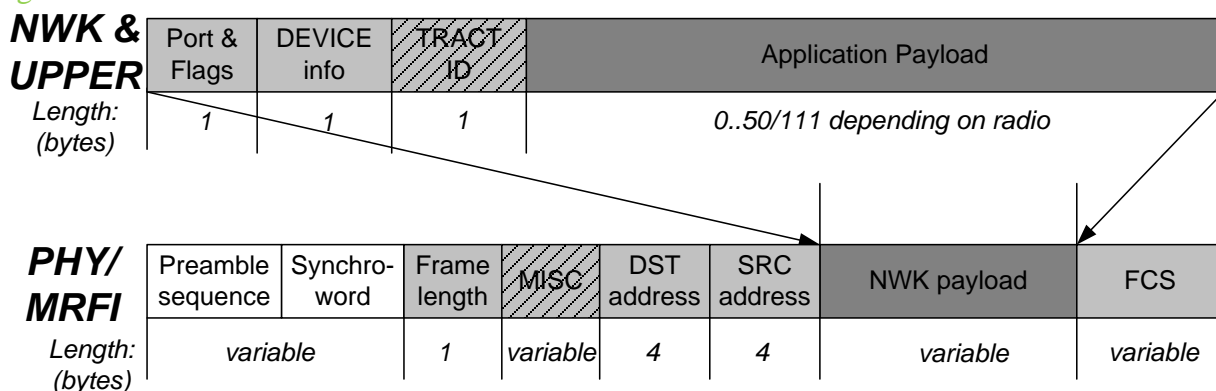


Fig. 1 b. Data frame format for Texas Instrument's SimpliciTI

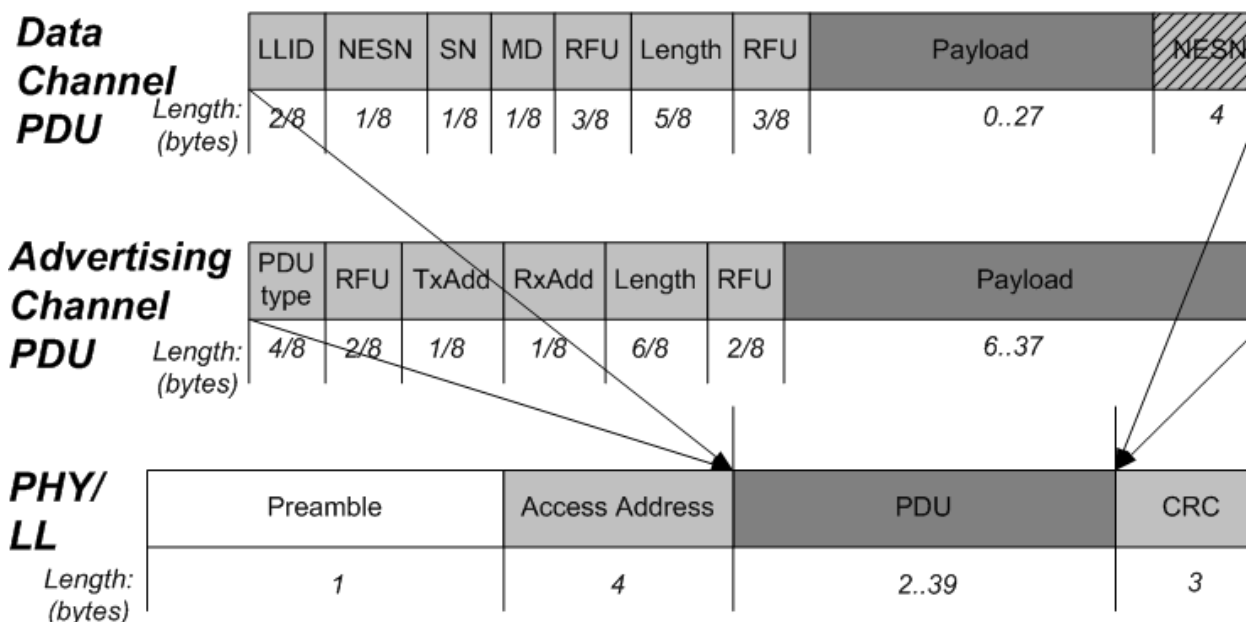


Fig. 1 c. Frame formats for BLE

### Proprietary protocols

Although the use of IEEE 802.15.4 provides some benefits for the WSNs, even now there are still many examples of when the WSN applications are implemented using other communication

protocols, quite often – the proprietary ones. The first and the most obvious reason for choosing this approach are the scenarios when the IEEE 802.15.4 cannot provide required performance. The good example for this scenario is the ANT protocol [34], [35] that uses Gaussian frequency-shift keying (GFSK) modulation with the high over-the-air rate (1000 kbit/s). This combination allows ANT to increase the data throughput (for the transmissions with a high duty cycle scenario) or the lifetime (for the low duty cycle scenario) at a cost of reduced communication range compared with the IEEE 802.15.4 [35]. The other reason for using the proprietary communication protocols can be the compatibility with the systems and applications designed before the IEEE 802.15.4 development. Finally, many of the radio chips based on the proprietary protocols (especially the ones utilizing FSK/GFSK or ASK modulation) have more simple structure and thus lower costs and power consumption than the radios implementing IEEE 802.15.4 [36]. Nonetheless, the drawback to this often is their lower communication range and higher error probability [37].

As the example of the proprietary protocol for this paper, we have chosen the open-source low-power radio protocol from Texas Instruments (TI) called SimpliciTI [38]. SimpliciTI has been developed by TI specifically for the wide range of their radio chips (both IEEE 802.15.4-compatible and proprietary ones) to enable fast and the low-cost development of the low-power wireless networking applications [38]. Like many other proprietary protocols, SimpliciTI lacks the rigid specification and is provided as the software (SW) stack with examples and minimum documentation required for SW development.

SimpliciTI do not define the PHY layer specifically - it is limited to the devices for which the stack support has been developed (TI CC1100/2500 – 433/868/2400 MHz band radio supporting OOK, FSK, GFSK, MSK; TI CC2420 – 2400 MHz band radio compatible with IEEE 802.15.4; and the chips, originating from those). The nominal over-the-air data rate for SimpliciTI is 250 kbit/s although it can be easily changed by the user as well as many other radio link settings. The structure of the data frame used by SimpliciTI is very similar to the one of IEEE 802.15.4 as shown in [Fig. 1b](#). The SimpliciTI frame begins with the sequence, which enables the bit- and byte-wise synchronization, and which is followed by the byte that specifies the length of the frame. After the frame length, the stack leaves space for some miscellaneous data (if required, e.g., for IEEE 802.15.4 radios can be inserted the frame control bytes [39][38]) and addresses of packet source and destination. Note that SimpliciTI does not use separate address fields for each layer and the 4-byte addresses are used by multiple layers [38]. The header for NWK and all upper layers in SimpliciTI occupies only three bytes. The first one (Port & Flags) is used for defining the target application and providing some information about the required payload processing (e.g., whether the payload should be decrypted before processing). The remaining bytes provide some additional data about sender and this packet (DEVICE ID) and the optional transaction ID (TRACT ID). The size of application payload for SimpliciTI ranges from 50 bytes (IEEE 802.15.4 non-compatible radios) and up to 111 bytes (IEEE 802.15.4 compatible radios) [39]. Together with the format of the NWK layer, SimpliciTI defines the so-called “applications” that can be used to manage the wireless network. The mandatory ones are: the *Ping* (detects the presence of a specific device), the *Link* (link management of two peers) and the *Mgm* (general management port for managing the device).

### Bluetooth Low Energy (BLE)

At the beginning of the century the IEEE 802.15.1 (Bluetooth) standard became very popular for the widest range of applications including, at some point, even the WSNs [40]. Nonetheless, the basic Bluetooth had many drawbacks that limited its application in WSNs. The most serious among those were: the utilized by Bluetooth communication model (master to slave), the limited number of the devices in a single cluster of the network (7 per a piconet), relatively long time required for a connection establishment (up to 33.28s), and low power efficiency [40], [41]. Those were the major reasons that caused the development of the BLE that is aimed specifically for providing the

very low cost and low energy consuming solution for short range low volume wireless data transferring [13].

The PHY layer of BLE is somewhat similar to the one of the classical Bluetooth v1: the BLE also operates in 2.4 GHz band and uses GFSK modulation (although the modulation index is increased from 0.35 to 0.5) with 1MBit/s over-the-air data rate and frequency hopping for spectrum spreading. Unlike the classical Bluetooth, which uses 79 1MHz-wide channels, the BLE uses 40 2-MHz wide channels (three of those that are located between Wireless LAN channels are used for advertising and service discovery, while the remaining 37 are utilized for data transmission). This solution is intended to reduce the connection establishment time and increase the power efficiency for the connection establishment.

The packet format defined by BLE link layer (LL) is presented in Fig. 1c. As Fig. 1c reveal the size of the packet and the maximum payload for BLE has been significantly reduced in comparison with the classical Bluetooth (see Table I). Each BLE packet starts with 1 byte preamble (classical Bluetooth utilizes 9-byte preamble), which is followed by the 4-byte access address – the random value that is unique to each LL connection between any two devices *on data channels or the specified constant on advertisement channels* [40]. After the access address, the packet contains the PDU, the format of which differs depending on the used channel, and the cyclic redundancy check (CRC) sequence calculated over the PDU. For the advertising channels, the PDU includes the 16-bit header with the data about the PDU type, some PDU specific information (TxAdd, RxAdd), the PDU size and some fields reserved for future use (RFU). The data channel besides the header and the payload the PDU can optionally contain the message integrity check (MIC) field if the LL connection is encrypted. The header of the data-channel PDU includes the fields of the logical link identifier (LLID), the next expected sequence number (NESN), the current sequence number (SN), additional data (i.e., “more data” (MD)), the length of the payload, and some reserved data.

The data exchange in BLE network can be implemented in two ways. The former one (see Fig. 2a) enables the transferring of all the data on the advertising channels. In this scenario, at a start of the advertising event the advertiser (i.e., the device that transmits advertising packets on the advertising channels) transmits the advertising packet that contains also the data itself. The other possible way for data transferring on the advertising channels is to send the request for required data. In this case, after receiving a data request, the advertiser that has the required data can use the same channel to send the required data back [42]. Note, that the channel, that is used for advertising is changed on the next advertising packet sent by the advertiser in the same advertising event [42]. This method can be used to send the data without the actual connection establishment procedures. Besides, it enables to include and use very simple BLE devices in the network that either only transmit or only receive the data.

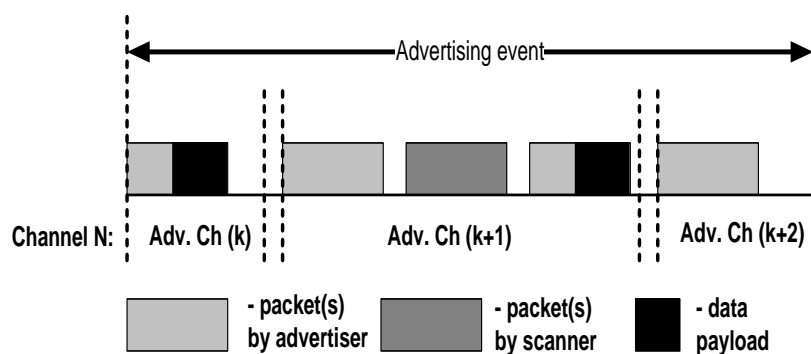


Fig. 2 a. Data transferring with BLE in advertising channels

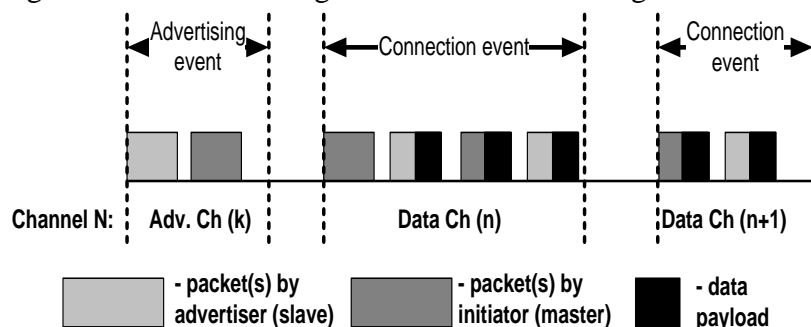


Fig. 2 b. Data transferring with BLE in data channels

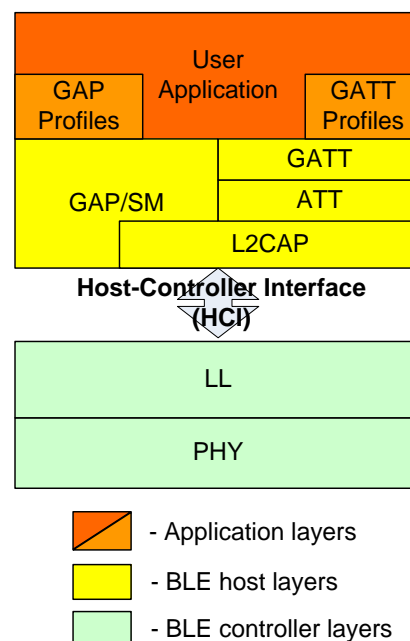


Fig. 3 Architecture of BLE protocol

The second option for BLE communication is to establish pair-wise duplex communication between two or more devices using data channels [42]. To form the connection (see Fig. 2b), the connection initiator (i.e., the device that needs to form a connection to another device) listens to advertising packets on the advertisement channels. Once detecting the advertisement packet from the required device (i.e., advertiser), the initiator may send the connection request over the same advertising channel. After that, if the advertiser receives and accepts the request for a connection, both devices start the communication using frequency hopping on the agreed data channels. During the data exchange, the initiator acts as the master unit and advertiser – as the slave. The master device initiates each connection event (i.e., the time unit which BLE devices can use for transmitting the data) and can end the connection at any time [42]. Note that the BLE only supports the connection between a master and a slave – the physical links between slaves are not supported. Moreover, in the current version of specification (see [42]) the slave device can be connected only to a single master device at a time and the role changes between a master and slave device are also prohibited.

The PHY and LL layers discussed above are the required components for each BLE controller device (i.e., the actual hardware device), nonetheless to comply the BLE specification the application should also use the required BLE host features (i.e., the specified upper layers that are implemented either by HW or by SW). According to [43] those include (see Fig. 3): L2CAP, GAP, ATT, GATT and SM. The logical link control and adaptation protocol (L2CAP) defines the procedures for higher level multiplexing, packet segmentation, and the conveying of quality of service (QoS) information. The generic access profile (GAP) specifies the generic procedures related to the discovery of devices, link establishment and termination management, procedures related to use of different security levels and also includes the common format requirements for parameters accessible on the user interface level. The security manager (SM) handles the management of pairing, authentication, bounding, and encryption for BLE communication. The attribute protocol (ATT) specifies the mechanisms for discovering, reading, and writing attributes on a peer device and the generic attribute profile (GATT) provides the framework for discovering services, and for reading and writing characteristic values on a peer device.

Table II: Nominal parameters and settings for the radio devices used in the tests

Parameter	Device		
	CC2510	CC2431	CC2540
Device type	system-on-chip (radio + 8051 microcontroller)	system-on-chip (radio + 8051 microcontroller)	system-on-chip (radio + 8051 microcontroller)
Microcontroller specification	Clock: 26 MHz Flash: 32 kB RAM: 4 kB	Clock: 32 MHz Flash: 128 kB RAM: 8 kB	Clock: 32 MHz Flash: 256 kB RAM: 8 kB
Radio protocol & stack	SimpliciTI (SimpliciTI)	IEEE 802.15.4 (TI-MAC)	BLE (TI BLE Stack)
Frequency band	2.4 GHz	2.4 GHz	2.4 GHz
Modulation	MSK	O-QPSK	GFSK
Spectrum spreading	None	DSSS	FHSS
Over-the-air data rate, kbit/s	250/500	250	1000
TX power range, dBm	-55...1	-25.2...0.6	-20...4
RX sensitivity, dBm	-90 (@250 kbit/s)	-92	-87 (standard mode)
Supply voltage, V	2-3.6	2-3.6	2-3.6
Sleep mode current consumption, $\mu$ A	0.5	0.5	0.9
Price (normalized)	0.358	1	0.341

Fig. 4 Hardware modules used for the tests: front row – extension radio boards (from left to right) CC2540 (BLE), CC2430 (IEEE 802.15.4) and CC2510 (FSK/SimpliciTI), back row – the battery extender (left) and the SmartRF04 development (right) boards



## Measurement methodology and results

### Test bed and measurement methodology.

To compare the performance of the protocols discussed in Section II in the real-life environment, we have used the CC2430, CC2510 and CC2540-based extension boards with SmartRF04 development boards from Texas Instruments (TI) (see Fig. 4). The main parameters of these radios are presented in Table 2. Our main targets in this study were:

- to define the maximum achievable application layer data rate for simplex single hop peer-to-peer (P2P) connection (SHP2PC) using the existing HW and SW;
- to compare the throughput of the protocols for simplex SHP2PC while using the packets of the same payload;
- to define and measure the minimum round trip time (RTT)<sup>2</sup> for SHP2PC;
- to compare the power consumed by the chips during communication;

<sup>2</sup> We have defined the minimum RTT as the period of time between the issue of the command to send the request data at the application layer and the reception of the response data at application layer on the same node, assuming that the node to which the request was sent does its best to response as fast as possible.



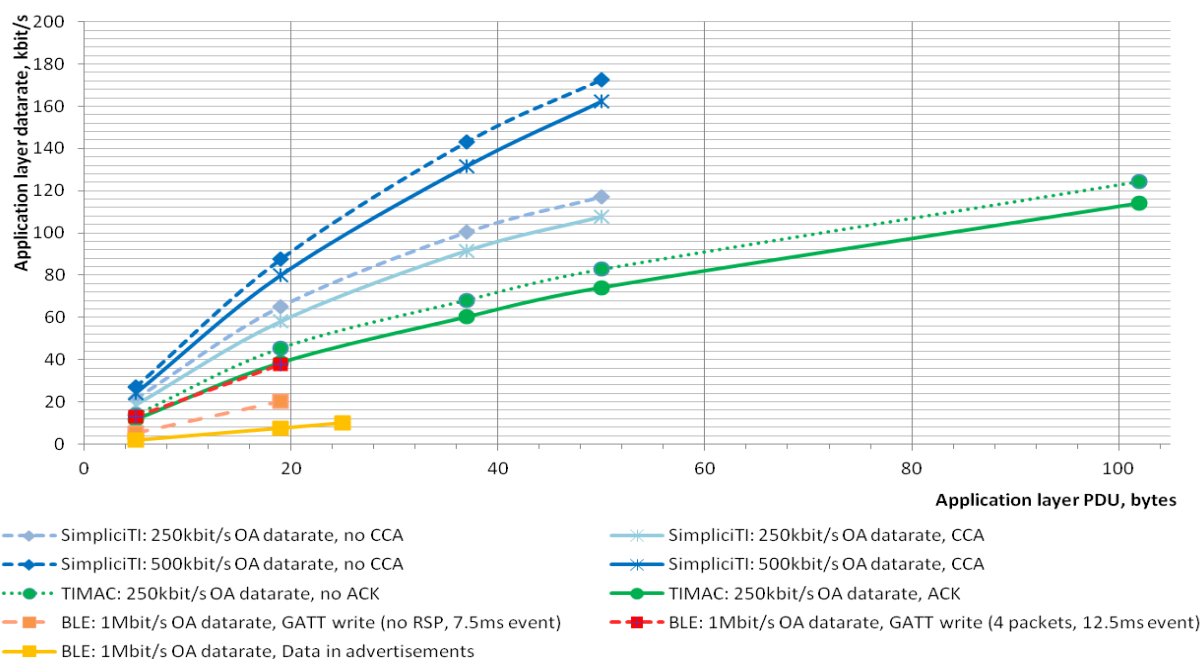


Fig. 5a Effect of application payload size on SHP2PC throughput

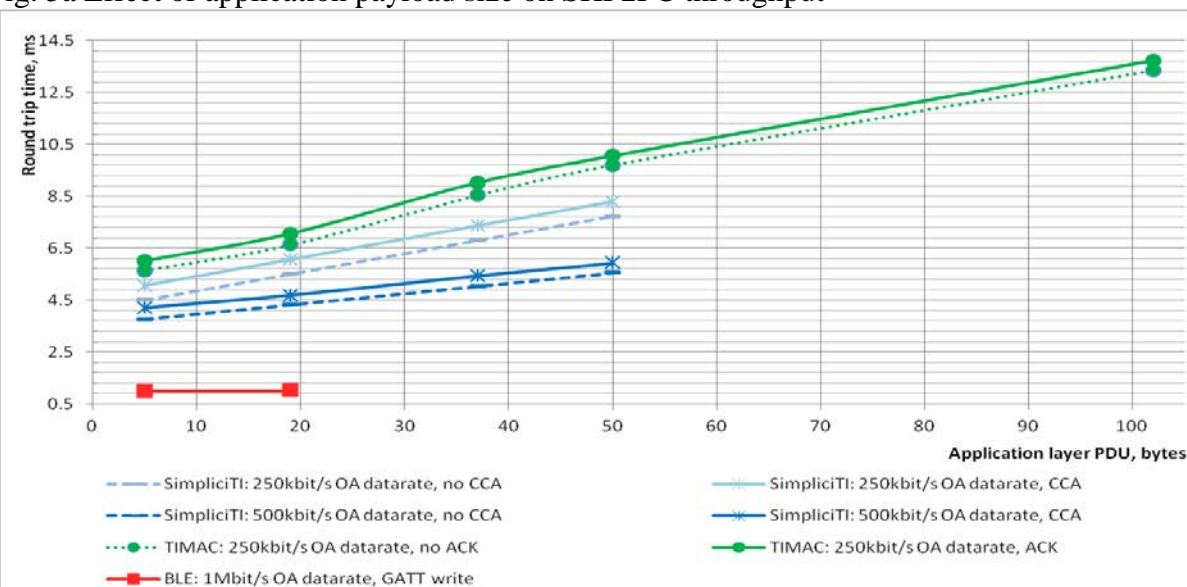


Fig. 5b Effect of application payload size<sup>1</sup> on SHP2PC RTT.

<sup>1</sup>-The payload of the specified size has been encapsulated both into the request and response packet.

- to compare the resources required for implementing the protocols.

During the tests, we have used SimpliciTI (version 1.2), TIMAC (version 1.0.1) and TI BLE (version 1.2.1) software stacks running on CC2510, CC2430 and CC2540 systems-on-chip (SoCs), respectively. Based on those stacks, we have developed the application code implementing the required measurement functionalities. The test data was injected by the developed application layers directly to the MAC layers implemented by SimpliciTI and TIMAC and to the GATT or GAP layers for BLE stack.

The measurements for the throughput and RTT were conducted in the laboratory environment using for SimpliciTI and 802.15.4 radios the channel with minimum interferences (the absence of interferences from other systems has been confirmed prior to the measurements). The measurements were executed with the transmitter board placed at a distance of around one meter from the receiver. This ensured that the strength of the radio signal for the receiver board was well

above the radio's sensitivity level (typically received signal strength indicator (RSSI) was well above -40 dBm) and enabled the test to have minimal packet losses (the packet error rate (PER) probability through the measurements was below 0.5%). Therefore, the data obtained from these measurements illustrate the best-case scenario. For decreasing the influence of the possible fluctuations, the throughput measurements have been executed for the sequence of 60,000 data packets and the RTT – with at least 1,000 packets sent forth and back (the time for connection re-establishing in the case of a packet loss is excluded from the resulting RTT).

For measuring the power consumption, we have used the current-shunt method (e.g., [44–46]). The expected maximum error for the conducted measurements is estimated to be below 3 mW. The power consumption measurements were executed for the radios supplied from the laboratory power source with the stabilized supply voltage of 3 V. During the power consumption measurements, all the peripherals on the test-boards were also switched off – the only active components on the board were the radio and the microcontroller core.

### **Measurement results**

As [Table II](#) reveals, all the hardware modules used in our test measurements have the same microcontroller core, which is based on the widespread 8051 microcontroller. This allowed us to have fair enough comparison for the complexity of the protocols by estimating the resources required for their implementation (see [Table III](#)). Note, that all the application programs used for comparing the resources had similar functionality and were compiled by the same version of integrated development environment (IDE) software with identical optimization settings. As possible to see from the presented results, the BLE stack required almost four times more program memory than TIMAC and more than seven times more resources than SimpliciTI. Although not all the resources have been consumed for the stacks directly (e.g., TIMAC and BLE stacks are implemented on top of OSAL operation system (OS), while SimpliciTI does not use any OS), we are sure that the presented data can indicate the magnitude and give the order of the complexity of the tested protocols.

Table III: Resource requirements for the stacks

Resource	Stack			
	SimpliciTI	Ti-MAC	BLE(Master)	BLE(Slave)
Program memory, Bytes	16 024	36 573	137 719	117 050
Data memory <sup>1</sup> , Bytes	3 567	5 438	12 750	10 676

<sup>1</sup> – cumulative for data in RAM and in Flash

Table IV: Power consumption for single packet transmission for tested radios

Stack	Required time, ms	Consumed energy, $\mu\text{J}$	Energy efficiency, $\mu\text{J}/\text{byte}$
<i>Transmission of a single packet with 19-byte user payload:</i>			
SimpliciTI, 250 kbit/s, CCA	2.46	165	8.7
SimpliciTI, 250 kbit/s, no CCA	2.21	148	7.8
SimpliciTI, 500 kbit/s, CCA	1.76	105	5.5
SimpliciTI, 500 kbit/s, no CCA	1.60	96	5.1
TIMAC, no ACK	1.74	135	7.1
BLE, GAP advertise <sup>1</sup>	0.88	47	2.5
BLE, GATT read <sup>2</sup>	0.83	45	2.4
<i>Transmission of data in a single packet with other payloads:</i>			
SimpliciTI, 50 bytes, 250 kbit/s, CCA	3.5	246	4.9
SimpliciTI, 50 bytes, 250 kbit/s, no CCA	3.16	227	4.5
SimpliciTI, 50 bytes, 500 kbit/s, CCA	2.23	148	3.0
SimpliciTI, 50 bytes, 500 kbit/s, no CCA	2.09	141	2.8
TIMAC, 102 bytes, no ACK	4.52	367	3.5
TIMAC, 102 bytes, ACK	5.17	419	4.0
BLE, GAP advertise <sup>1</sup> 26bytes	0.88	63	2.4

<sup>1</sup> – The energy consumption for checking the channel for possible connection request after advertisement is accounted (see Fig. 6a)

<sup>2</sup> – The data are sent after the reception of request. The energy consumption for request reception is accounted (see, e.g., Fig. 6a)

The performance test results of the different protocols are presented in Figs. 5a and 5b. As Fig 5a shows the maximum SHP2PC application layer data rate obtained for tested BLE radio and stack was around 37 kbit/s. This data rate was achieved while setting the BLE connection event *interval* in the stack to 12.5 ms with four data packets with maximum payload (maximum user payload while using GATT and TI BLE stack is 19 bytes) transmitted within each connection event (in reality – quite often instead of four only three packets were sent and the length of the connection event was slightly bigger). Note that the BLE DL layer of the receiving node automatically acknowledged each received packet. Nonetheless, we assume that the major reason for such poor throughput result for BLE radios was not the features of the protocol itself, but the implementation limitations to the TI BLE stack. At the current point, the TI BLE stack artificially limits the maximum amount of data transmitted within a single connection event to 80 bytes, or 4 full-size packets, which had dramatically reduced the maximum achievable SHP2PC throughput. As possible to expect based on the results presented in Figs. 5b and 6a, without this limitation, the maximum SHP2PC data rate for BLE radios could reach 200 kbit/s (packet with 19 bytes of payload transmitted every 0.7-0.8 ms)

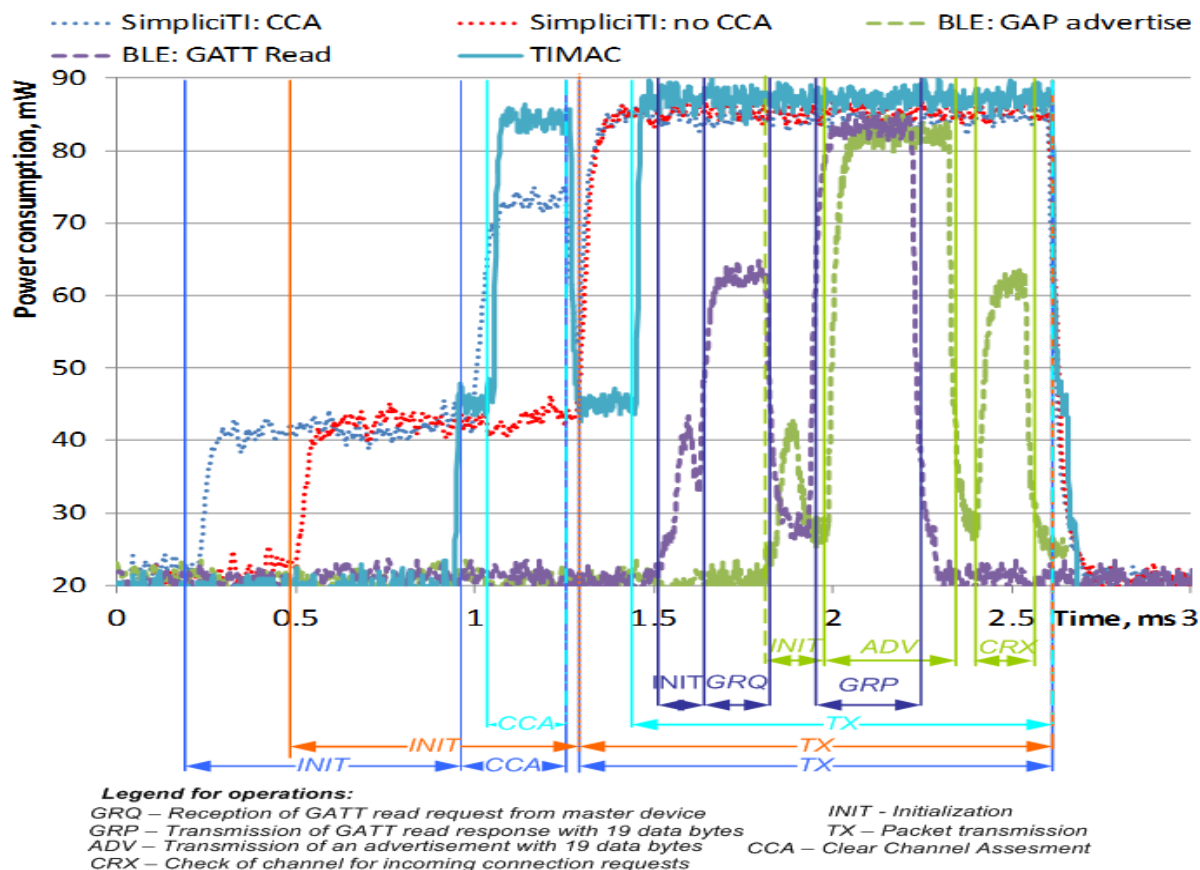


Fig. 6a Power consumption by different radios and protocols during different transmission stages of a packet with 19-byte user-defined payload (supply voltage 3 V, radio transmit power 0 dBm)

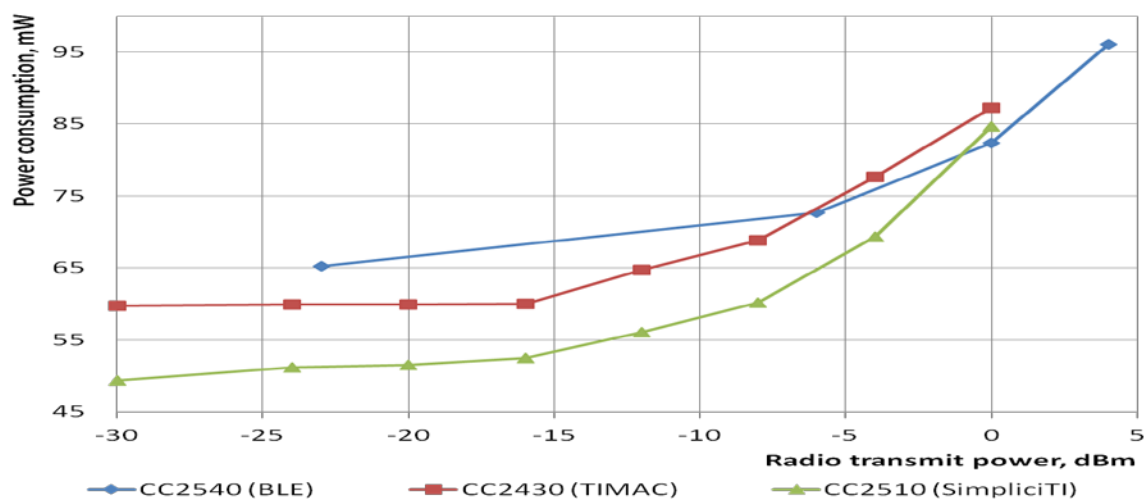


Fig. 6b Effect of the transmit power on the power consumption for tested radios (supply voltage 3 V)

A slightly lower data rate (about 10 kbit/s) we have obtained with BLE while encapsulating application data directly into the advertisement packets (maximum user-defined data equal to 25 bytes were encapsulated into advertisements transmitted every 20 ms (minimum period according to [1])). Nonetheless, empirically has been proved that the current versions of HW and SW are actually capable of sending the advertisements with the period of 6.8 ms which could allow to get the total application data rate of around 28 kbit/s. The advantage of this method is that the data are

transmitted on the advertisement channel and therefore can be received by multiple devices without any connection establishment procedure, the drawbacks to it are: the impossibility of using the acknowledgements; complexity of using security techniques and the possibility of obstructions for other BLE devices due to very limited advertisement channels' bandwidths. As possible to see from Fig. 5a, the data rate obtained using BLE with four packets per event is very close to the data rate for TIMAC radio operating in beacon-disabled mode with automatic hardware acknowledgements (ACKs). Although the over-the-air data rate for CC2430 radios is four times lower than for BLE, these radios can support much longer payload packets and do not require up-keeping the synchronization. This allowed obtaining the maximum SHP2PC data rate for CC2430 of 124 kbit/s with ACKs and 114 kbit/s without ACKs. Using CC2510 radio with SimpliciTI stack we managed to obtain the maximum application-layer data rate of 117 kbit/s and 172 kbit/s (50 bytes payload, CCA disabled) for 250 kbit/s and 500 kbit/s over-the-air data rate, respectively. Nonetheless, unlike CC2540 and CC2430, CC2510 has no support for hardware packet acknowledgement. This will result in the significant reduction of the total throughput if the acknowledgement mechanism will be implemented at the application layer (see Fig. 5b).

As Fig. 5b reveals the RTT for BLE radios (once the connection is established and if both transmissions are executed within the same connection event) appeared to be significantly lower than for the other radios. Note, that the measurements for RTT for BLE were done using two GATT write packets: firstly the master device was writing some value to attribute table of the slave, then vice versa.

The results illustrating the power consumption of the different radios are presented in Figs. 6a and 6b and Table IV. The Fig. 6a showing the power consumption profiles for the tested protocols and radios during 19-byte packet transmission reveals that BLE radios required much less time and energy for the actual data transmission than the other protocols. The major reasons for that are the high over-the-air data rate used by BLE, the absence of CCA before packet transmission, and low overhead. Nonetheless, Fig. 6a depicts that in both tested scenarios the BLE also spent some time in receive: to catch the GATT request over the data channels or to check for incoming connection request while using advertisement channels. As revealed in Table IV, which summarizes the power consumption for data transmission for the protocols discussed while transferring the same amount of data (i.e., 19 bytes), the BLE radios required two to three times less energy compared with the other tested protocols. This result corresponds quite well to the results presented in [29]. When the energy consumption of the BLE is compared with the other protocols using higher packet payloads (see Table IV) we note that even in that case the amount of energy required for transferring the data with BLE is lower. An interesting conclusion can be drawn from Fig. 6b that reveals the effect of the radio transmit power level on the total power consumption of the device during the transmission period: although the BLE consumed less power than the other radios for sending the data at power level of 0dBm, for sending the signal at power level of -6 dBm and below BLE required more power than other tested radios. This allows us to expect that for the transmit signal power level below -20 dBm the energy efficiency of the BLE radio will be comparable at least with the SimpliciTI radios using maximum payload (see Table IV and Fig. 6b).

Note, that the presented measurements reveal the best-case scenario, when there were almost no interferences. Meanwhile, in the real-life scenario multiple devices can operate close to each other and some packets can get lost due to the interferences. The retransmission of such packets will affect both the obtainable data rate and the energy consumption. Although the packet losses can happen for all discussed protocols, we suppose that for BLE this can give especially strong effect due to the absence of CCA mechanism before packet retransmission, the lower number of the available channels and the absence of adaptive frequency hopping mechanism in some of the current BLE stack implementations (see, e.g. [29]). Nonetheless, further research is required to confirm or to refute this.

## Discussion and conclusions

The Bluetooth Low Energy (BLE) is relatively new protocol and this study is among the first publications that contribute providing the results for the performance and power consumption measurement achievable using the real-life hardware radios and software stacks. The results of the measurements for BLE have been compared with those of two other protocols: the TIMAC implementing the IEEE 802.15.4, which can be considered as the de-facto standard for WSNs nowadays and SimpliciTI – the proprietary and very simple protocol developed by Texas Instruments. We would like to note, the presented results are intended to reveal some main features of the tested radios, for the developed test programs based on the currently available protocol stack releases, instead of representing the ultimate performance and power consumption values for the stacks.

The presented results reveal that the BLE radios had significantly lower energy consumption than the other tested radios and protocols, which is due to the higher over-the-air data rate and absence of carrier sense mechanism of BLE. Besides, the same features discussed above allow BLE to support very low RTT – the receiver radio was capable of receiving the request, processing it, and sending the response packet back within less than 1 ms. Nonetheless, the implementation limitations in the used version of the BLE software stack dramatically reduced the maximum achievable data rate on data channel and made it much lower than the one available for IEEE 802.15.4 or SimpliciTI radios. The maximum data rate obtained on the advertisement channels was several times lower than the data rate on the data channels.

Another restriction of application areas, which we expect for BLE, is its lower maximum communication range compared with the other protocols. It is well known (as shown e.g., in Fig. 10.1 in [47] or in Fig. 7.11 in [48]), that the radio with FSK modulation has to use higher transmit power to get the same probability of error as the PSK radio. Besides, the use of higher over-the-air data rate typically reduces the sensitivity of the receiver (see [Table II](#)). Those two factors will cause the BLE radios to have the lower maximum communication range than the other protocols for the same transmission power value. Partially, this can be solved by using the higher level of the transmit power, as, e.g., the tested BLE radios are capable of transmitting the signal at power level of 4 dBm, while the radios implementing other protocols can give only 1 dBm at the best. Nonetheless, this approach obviously causes the increase in the power consumption for the transmitter node and might be impossible in some regions due to radio spectrum usage regulations. Another very serious limitation of the BLE protocol especially for the WSN applications is the networking topology restrictions which require each slave to be connected only to one master node at a time and allow only single-hop connections between master and slave nodes.

Nonetheless, BLE has two very important advantages: the interoperability and the low cost. Indeed, to obtain the data from a WSN nowadays the user has to use the external hardware radio module attached to his smartphone or personal computer. Meanwhile, the first smart phones encapsulating the dual mode Bluetooth (standard Bluetooth + BLE) radio chip have already appeared on the market. In future this will enable to significantly simplify and generalize the process of establishing the connection between the BLE sensors or actuators and the user devices and will boost for many new applications, including the WSN ones. Besides, as has been shown in the paper, out of all tested radios the BLE ones have the lowest cost (which is quite surprising as the microcontroller core of the BLE radio required much more resources for implementing the stack than are available on other radios).

Therefore, we assume that BLE can find the wide application area in WSNs although the estimation of BLE dominating the market by 2015 expected by authors in [2] is hardly be fulfilled. Based on the presented results, we expect that the major application area for BLE at least in the nearest years is going to be the energy efficient human-oriented applications with single-hop topology. Those can include: health and fitness; entertainment; smart home; personal security and proximity detection; data and location advertisement. Meanwhile, we suppose that for the non-human oriented applications, for professional application and for the applications requiring high coverage

area (e.g., the wildlife, nature and environment monitoring, industrial monitoring and control, building and process automation and security, logistics) will still be widely used the other communication protocols and standards such as IEEE 802.15.4.

### References:

- [1] "Core System Package [Low Energy Controller volume]," *BLUETOOTH SPECIFICATION Version 4.0*, vol. 6, pp. 1–128, 2010.
- [2] West Technologies Research Solutions, "WTRS Wireless Sensor Network Technology Trends Report," WT062510CNTS, Jun. 2010.
- [3] H. Fornazier, A. Martin, and S. Messner, "Wireless Communication : Wi-Fi, Bluetooth, IEEE 802.15.4, DASH7," 2012.
- [4] DASH7 Alliance, "DASH7TM Technical Overview Webinar," 02-Dec-2009.
- [5] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *in Proceedings of IECON'07*, 2007, pp. 46–51.
- [6] N. Ullah, M. S. Chowdhury, P. Khan, and K. S. Kwak, "Throughput limits of UWB based 802.15.4a," in *in Proceedings of ICTC'10*, 2010, pp. 166–167.
- [7] Z. Ahmadian and L. Lampe, "Performance Analysis of the IEEE 802.15.4a UWB System," *IEEE Transactions on Communications*, vol. 57, no. 5, pp. 1474–1485, May 2009.
- [8] F. Haroon, K. M. Ahmed, and H. Rasheed, "An adaption of IEEE 802.15.4a in an industrial Field Bus," in *in Proceedings of INDIN'10*, 2010, pp. 531–536.
- [9] L. De Nardis and M.-G. Di Benedetto, "Overview of the IEEE 802.15.4/4a standards for low data rate Wireless Personal Data Networks," in *in Proceedings of WPNC'07*, 2007, pp. 285–289.
- [10] E. Karapistoli, F.-N. Pavlidou, I. Gragopoulos, and I. Tsetsinas, "An overview of the IEEE 802.15.4a Standard," *IEEE Communications Magazine*, vol. 48, no. 1, pp. 47–53, Jan. 2010.
- [11] H. J. Rivertz, "Bluetooth Security," DART/05/05, Mar. 2005.
- [12] DASH7 Alliance, "OpenTag: Office Hours," 12-May-2011.
- [13] LitePoint Corporation, "Bluetooth® Low Energy," Doc: 1075-0019-001, LitePoint Corporation.
- [14] A. J. Jara, D. Fern'andez, P. Lopez, M. A. Zamora, B. Ubeda, and A. G. Skarmeta, "Evaluation of Bluetooth Low Energy Capabilities for Continuous Data Transmission from a Wearable Electrocardiogram," in *in Proceedings of IMIS'12*, 2012, pp. 912–917.
- [15] C. Gomez, J. Oller, and J. Paradells, "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology," *Sensors*, vol. 12, no. 9, pp. 11734–11753, Aug. 2012.
- [16] "Parameters for active air interface communications at 433 MHz," *ISO/IEC 18000-7:2004(E)*, pp. 1–28, 2004.
- [17] "Wireless Short-Packet (WSP) protocol optimized for energy harvesting — Architecture and lower layer protocols," *ISO/IEC DIS 14543-3-10*, pp. 1–31, 2011.
- [18] "Short range narrow-band digital radiocommunication transceivers – PHY and MAC layer specifications," *ITU-T G.9959*, pp. 1–126, 2012.
- [19] M. Park, "Proposed Specification Framework for TGah," IEEE 802.11-11/1137r11, Sep. 2012.
- [20] K. Kwak, S. Ullah, and N. Ullah, "An overview of IEEE 802.15.6 standard," in *in Proceedings ISABEL'10*, 2010, pp. 1–6.
- [21] "IEEE Standard for Local and metropolitan area networks - Part 15.6: Wireless Body Area Networks," *IEEE Std 802.15.6-2012*, pp. 1–271, 2012.
- [22] Y. Zhang, A. Atac, L. Liao, and S. Heinen, "A low-power high-efficiency demodulator in bluetooth low energy receiver," in *in Proceedings PRIME'12*, 2012, pp. 1–4.
- [23] J. Masuch and M. Delgado-Restituto, "A 190-microWatt zero-IF GFSK Demodulator With a 4-b Phase-Domain ADC," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 11, pp. 2796–2806, Nov. 2012.
- [24] A. Wong, M. Dawkins, G. Devita, N. Kasparidis, A. Katsiamis, O. King, F. Lauria, J. Schiff, and A. Burdett, "A 1V 5mA multimode IEEE 802.15.6/bluetooth low-energy WBAN transceiver for biotelemetry applications," in *in Proceedings of ISSCC'12*, 2012, pp. 300–302.
- [25] B. Yu, L. Xu, and Y. Li, "Bluetooth Low Energy (BLE) based mobile electrocardiogram monitoring system," in *in Proceedings ICIA'12*, 2012, pp. 763–767.
- [26] M. Ali, L. Albasha, and H. Al-Nashash, "A Bluetooth low energy implantable glucose monitoring system," in *in Proceedings EuRAD'11*, 2011, pp. 377–380.
- [27] J. Liu, C. Chen, and Y. Ma, "Modeling Neighbor Discovery in Bluetooth Low Energy Networks," *IEEE Communications Letters*, vol. 16, no. 9, pp. 1439–1441, Sep. 2012.
- [28] S. Kamath and J. Lindh, "Measuring Bluetooth® Low Energy Power Consumption," Texas Instruments, Inc, AN092S (WRA347a), 2012.

- [29] M. Siekkinen, M. Hienkari, J. K. Nurminen, and J. Nieminen, "How low energy is bluetooth low energy? Comparative measurements with ZigBee/802.15.4," in *in Proceedings WCNCW'12*, 2012, pp. 232–237.
- [30] "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, 2011.
- [31] E. P. Capo-Chichi, H. Guyennet, and J.-M. Friedt, "IEEE 802.15.4 Performance on a Hierarchical Hybrid Sensor Network Platform," in *in Proceedings ICNS'09*, 2009, pp. 303–308.
- [32] C. Buratti, M. Martalo, R. Verdone, and G. Ferrari, *Sensor Networks with IEEE 802.15.4 Systems*, 1st ed. Berlin Heidelberg: Springer, 2011.
- [33] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)," *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pp. 1–320, 2006.
- [34] N. Sheriff, "Time Synchronization in ANT Wireless Low Power Sensor Network," M. Sc. Thesis, Jönköping University, Jönköping, Sweden, 2010.
- [35] R. Morris, "Exceeding the standard for wireless sensor networks," *EE Times-Asia*, pp. 1–4, May-2008.
- [36] W. Rahajandraibe, L. Zaid, V. Cheynet de Beaupre, and G. Bas, "Frequency Synthesizer and FSK Modulator for IEEE 802.15.4 Based Applications," in *in Proceedings of RFIC'07*, 2007, pp. 229–232.
- [37] J. T. Adams, "ZIGBEE WIRELESS TECHNOLOGY AND THE IEEE 802.15.4 RADIO – ENABLING SIMPLE WIRELESS," in *Proceedings of Texas Wireless Symposium'05*, 2005, pp. 76–80.
- [38] Texas Instruments, "SimpliciTI™." .
- [39] L. Friedman, "SimpliciTI: Simple Modular RF Network Specification," Texas Instruments, Inc, 2009.
- [40] S. Krco, "Bluetooth Based Wireless Sensor Networks –Implementation Issues and Solutions," in *Proceedings of TELFOR'02*, Belgrade, Serbia, 2002, p. 7.
- [41] R. Heydon, *Bluetooth Low Energy: The Developer's Handbook*. Upper Saddle River, New Jersey: Prentice Hall, 2012.
- [42] "Core System Package [Architecture & Terminology Overview]," *BLUETOOTH SPECIFICATION Version 4.0*, vol. 1, pp. 1–140, 2010.
- [43] "Core System Package [Master Table of Contents & Compliance Requirements]," *BLUETOOTH SPECIFICATION Version 4.0*, vol. 0, pp. 1–106, 2010.
- [44] Z. Nakutis, "Embedded systems power consumption measurement methods overview," *MATAVIMAI*, vol. 2, no. 44, pp. 29–35, 2009.
- [45] K. Mikhaylov and J. Tervonen, "Optimization of microcontroller hardware parameters for Wireless Sensor Network node power consumption and lifetime improvement," in *Proceedings of ICUMT'10*, Moscow, Russia, 2010, pp. 1150–1156.
- [46] K. Mikhaylov and J. Tervonen, "Evaluation of Power Efficiency for Digital Serial Interfaces of Microcontrollers," in *Proceedings of NTMS'12*, Istanbul, Turkey, 2012, pp. 1–5.
- [47] K. V. K. K. Prasad, *Principles of Digital Communication Systems and Computer Networks*. Cengage Learning, 2004.
- [48] K. Pahlavan and A. H. Levesque, *Wireless Information Networks*. John Wiley & Sons, 2005.



## ΠΑΡΑΡΤΗΜΑ 2

Στις παρακάτω σελίδες παρουσιάζεται ο πηγαίος κώδικας που αναπτύχθηκε στο C8051F020-DK αναπτυξιακό και χρησιμοποιήθηκε ως γεννήτρια προκαθορισμένων σειριακών δεδομένων κατά βούληση από τον χρήστη με push button.

```
#include <ioC8051f020.h>           // SFR declarations
#include <stdio.h>
#include <intrinsics.h>
#include <stdbool.h>
#include <pgmspace.h>

#define nop()  (__no_operation())

//-----
// Global CONSTANTS
//-----

#define BAUDRATE  9600 //38400      // Baud rate of UART in bps
#define SYSCLK    22118400        // SYSCLK frequency in Hz

//#define INT_DEC  256             // integrate and decimate ratio

#define HW_RTS    P4_bit.P40      // Ready to Send bit
#define HW_CTS    P4_bit.P41      // Clear to Send bit

#define SW_CTS    P0_bit.P02      //CTS soft implementation

#define LED P1_bit.P16            // LED='1' means ON
#define SW1 P3_bit.P37           // SW1='0' means switch pressed
#define ES0 IE_bit.ES            // enable uart0 interrupt

#define EA IE_bit.EA             // Global Enable ISRs
#define TR1 TCON_bit.TR1

#define TI0 SCON_bit.TI          // TX0 ISR flag
#define RI0 SCON_bit.RI          // RX0 ISR flag

/*
1. before transmission, assert RTS
2. wait for CTS to be asserted by the other device
3. when CTS is asserted, start transmission
4. when finished transmitting, de-assert RTS
*/

typedef unsigned int uint16_t;
typedef unsigned char uint8_t;
typedef unsigned long uint32_t;

//-----
```

```

// Function PROTOTYPES
//-----

void SYSCLK_Init (void);
void PORT_Init (void);
void UART0_Init (void);
__interrupt void UART0_RX_ISR (void);

//-----
// Global VARIABLES
//-----

__data uint16_t TXdata_c;
__data uint16_t RXdata_c;

//__data uint8_t chsmtable[10]={'0','1','2','3','4','5','6','7','8','9'};
__data uint8_t TX_char='0';

__no_init __bit bool sw1_press_show_b;
__no_init __bit bool sw1_debunce_b;
__no_init __bit bool TX_stop_b;
__no_init __bit bool no_delay_b;

//-----
// MAIN Routine
//-----
static void putc2serial (register uint8_t x)
{
    //EA = false;
    SBUF0 = x;
    //TIO = 0;
    // EA = true;
    TXdata_c++;
    while(!TIO);
    TIO = 0;
}

void main (void)
{
    WDTCN = 0xde;           // disable watchdog timer
    WDTCN = 0xad;

    SYSCLK_Init();        // initialize oscillator
    PORT_Init();          // initialize crossbar and GPIO
    UART0_Init();         // initialize UART0
    TIO = false;          // no isr used
    LED = false;
    TX_stop_b = false;
    ES0 = 1;              // Enable serial0 ISR
    no_delay_b = false;
    sw1_press_show_b = true;
}

```

```

sw1_debounce_b = true;
EA = 1;                                // Enable global interrupts

uint16_t x=0;
uint16_t y=0;

while(1)
{

if(!SW1)
{

if(sw1_debounce_b == true)
{

for(uint16_t t=0; t<20000; t++)
nop();

sw1_debounce_b = false;
}

#ifdef TX_FLOW_CONTROL
while(SW_CTS == true);                //wait the receiver to be ready (Soft TX Flow Control)
#endif
if(TX_char>'8')
{
if(sw1_press_show_b==true)
{
putc2serial('*');
sw1_press_show_b = false;
}
else
{
putc2serial(' ');
TX_char='0';
}
}
else
{
if(sw1_press_show_b==true)
{
putc2serial('*');
sw1_press_show_b = false;
}
else
{
putc2serial(++TX_char);              // code for transmit
}
}

}
else

```



```

for (i=0; i<1000; i++);           // XTLVLD blanking interval (>1ms)

while (!(OSCCCN & 0x80));         // Wait for crystal osc. to settle

OSCCCN = 0x88;                   // select external oscillator as SYSCLK
                                  // source and enable missing clock
                                  // detector
}

//-----
// PORT_Init
//-----
//
// Configure the Crossbar and GPIO ports
//
void PORT_Init (void)
{
    XBR0 = 0x04;                   // Enable UART0
    XBR1 = 0x00;
    XBR2 = 0x40;                   // Enable crossbar and weak pull-ups
    POMDOUT |= 0x01;              // enable TX0 as a push-pull output
    P1MDOUT |= 0x40;              // enable P1.6 (LED) as push-pull output
}

//-----
// UART0_Init
//-----
//
// Configure the UART0 using Timer1, for <baudrate> and 8-N-1.
//
void UART0_Init (void)
{
    SCON = 0x50;                   // SCON0: mode 1, 8-bit UART, enable RX
    TMOD = 0x20;                   // TMOD: timer 1, mode 2, 8-bit reload
    TH1 = -(SYSCLK/BAUDRATE/16);  // set Timer1 reload value for baudrate
    TR1 = 1;                       // start Timer1
    CKCON |= 0x10;                 // Timer1 uses SYSCLK as time base
    PCON |= 0x80;                  // SMOD00 = 1
    TIO = 1;                       // Indicate TX0 ready
}

#pragma vector=0x23 // interrupt 4
__interrupt void UART0_RX_ISR (void)
{
    if(RIO == true)
    {
        RIO = 0;
        RXdata_c++;
    }
}

```