

Εργαστήριο 1: Εισαγωγή στο Software Defined Radio

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Πανεπιστήμιο Δυτικής Μακεδονίας

Μάρκος Δελαπόρτας¹

Μάρτιος 2024

¹E-mail: ece01316@uowm.gr

Abstract

Ένα Software Defined Radio (SDR) είναι ένα ραδιόφωνο που χρησιμοποιεί λογισμικό για την υλοποίηση των αλγορίθμων που είναι απαραίτητοι για την ψηφιακή επικοινωνία. Σε αυτό το εργαστήριο, θα σχεδιάζετε και θα υλοποιείτε ένα SDR χρησιμοποιώντας το Universal Software Radio Peripheral (USRP) και το λογισμικό GNU Radio. Ο σκοπός αυτής της εισαγωγικής εργαστηριακής άσκησης είναι να εξασφαλιστεί ότι οι μαθητές έχουν μια λειτουργική εγκατάσταση του GNU Radio στους υπολογιστές τους και γνωρίζουν πώς να συνδεθούν με το λογισμικό του USRP.

I Εισαγωγή

Το Wireless Innovation Forum (WINNF) ορίζει το Software Defined Radio ως: "Ραδιόφωνο στο οποίο ορισμένες ή όλες οι λειτουργίες του φυσικού επιπέδου καθορίζονται από το λογισμικό". Το SDR αναφέρεται στην τεχνολογία όπου οι μονάδες λογισμικού που εκτελούνται σε μια γενική πλατφόρμα υλικού χρησιμοποιούνται για την υλοποίηση ραδιολειτουργιών. Συνδυάζοντας το υλικό USRP με το λογισμικό GNU Radio μπορείτε να δημιουργήσετε μια ευέλικτη και λειτουργική πλατφόρμα SDR για γρήγορη δημιουργία πρωτοτύπων ασύρματων σημάτων, συμπεριλαμβανομένου του σχεδιασμού φυσικών επιπέδων, της εγγραφής και αναπαραγωγής, της ευφυΐας σήματος, της επικύρωσης αλγορίθμων και άλλων.

I.I USRP Hardware Features

Τα ραδιοφωνικά προϊόντα Universal Software Radio Peripheral (USRP) έχουν σχεδιαστεί για εφαρμογές ραδιοσυχνοτήτων από DC έως 6 GHz, συμπεριλαμβανομένων συστημάτων πολλαπλών κεραιών (MIMO). Παραδείγματα τομέων εφαρμογής περιλαμβάνουν ανηχοϊκά δωμάτια, κινητά τηλέφωνα, παρακολούθηση φάσματος, ραδιοδικτύωση, γνωστικό ραδιόφωνο, δορυφορική πλοήγηση και ερασιτεχνικό ραδιόφωνο. Το USRP συνδέεται με έναν υπολογιστή δημιουργώντας έτσι ένα ραδιόφωνο καθορισμένο από λογισμικό.

Για τη Λήψη

- Τα εισερχόμενα σήματα στις εισόδους της σύνδεσης SMA αναμιγνύονται χρησιμοποιώντας έναν δέκτη άμεσης μετατροπής σε στοιχεία I/Q (in-phase/quadrature) βασικής ζώνης.
- Η δειγματοληψία των δεδομένων I/Q θα γίνει από έναν αναλογικό σε ψηφιακό μετατροπέα (ADC).
- Τα ψηφιοποιημένα δεδομένα I/Q ακολουθούν παράλληλες διαδρομές μέσω μιας διαδικασίας digital down-conversion (DDC) που αναμιγνύει, φιλτράρει και κατακερματίζει το σήμα εισόδου σε ρυθμό καθορισμένο από το χρήστη.
- Τα κατακερματισμένα δείγματα μεταβιβάζονται στον κεντρικό υπολογιστή.

Για τη Μετάδοση

- Τα δείγματα I/Q σήματος βασικής ζώνης συντίθενται από τον κεντρικό υπολογιστή και τροφοδοτούνται στο USRP με καθορισμένο ρυθμό δειγματοληψίας μέσω USB.
- Το υλικό USRP παρεμβάλλει το εισερχόμενο σήμα σε υψηλότερο ρυθμό δειγματοληψίας χρησιμοποιώντας μια διαδικασία digital up-conversion (DUC).
- Στη συνέχεια, μετατρέπει το σήμα σε αναλογικό με έναν ψηφιακό σε αναλογικό μετατροπέα (DAC).
- Το προκύπτον αναλογικό σήμα στη συνέχεια διαμορφώνεται με την καθορισμένη συχνότητα φορέα και μεταδίδεται μέσω των συνδέσμων SMA.

Οι πράξεις διαβίβασης ή λήψης συνοψίζονται στο σχήμα (1)

Στο εργαστήριό μας, θα χρησιμοποιήσουμε το υλικό USRP B200 και B210, δείτε τα σχήματα (2b), (2a) και (2c). Η σειρά B(us) USRP καλύπτει συχνότητες RF από 70 MHz έως 6 GHz, διαθέτει συνδεσιμότητα Spartan-6 FPGA και USB 3.0. Αυτή η πλατφόρμα επιτρέπει τον πειραματισμό με ένα ευρύ φάσμα σημάτων, συμπεριλαμβανομένων FM και τηλεοπτικών εκπομπών, κινητής τηλεφωνίας, Wi-Fi και άλλων. Το USRP B200 διαθέτει ένα κανάλι λήψης και ένα κανάλι μετάδοσης σε σχεδιασμό που τροφοδοτείται από δίαυλο. Το USRP B210 επεκτείνει τις δυνατότητες του

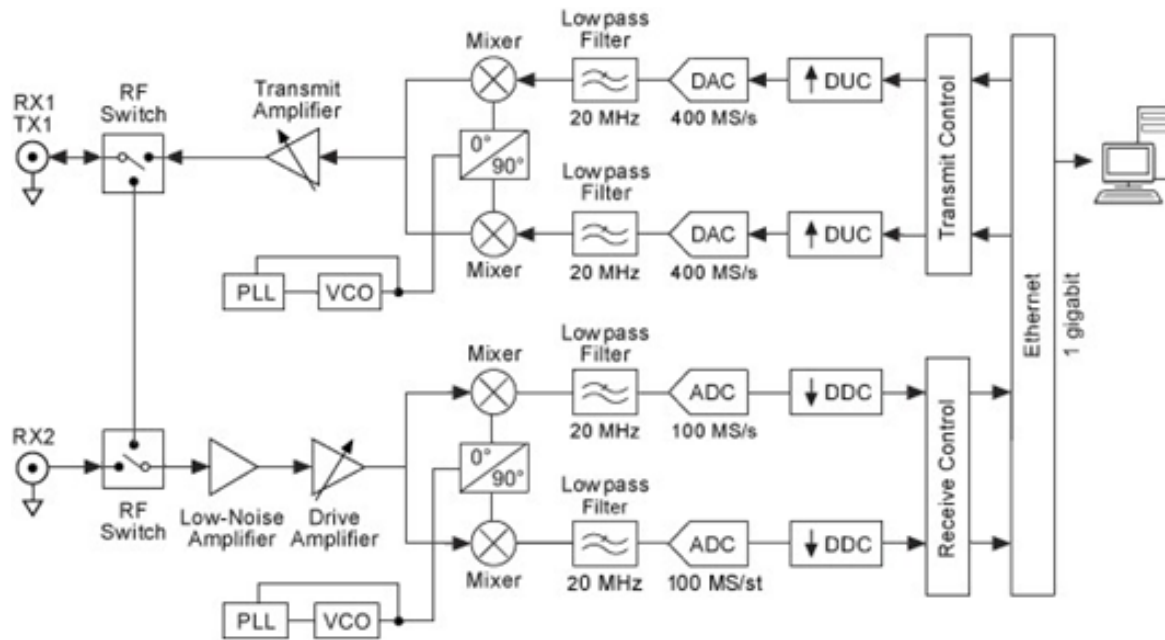


Figure 1: Τυπικό μπλοκ διάγραμμα ενός USRP

B200 προσφέροντας συνολικά δύο κανάλια λήψης και δύο κανάλια μετάδοσης, ενσωματώνει ένα μεγαλύτερο FPGA, GPIO και περιλαμβάνει εξωτερικό τροφοδοτικό. Και οι δύο χρησιμοποιούν ένα RFIC αναλογικών συσκευών (AD9361) για να προσφέρουν μια οικονομικά αποδοτική πλατφόρμα πειραματισμού RF και μπορούν να μεταδώσουν έως και 56 MHz στιγμιαίου εύρους ζώνης μέσω ενός διαύλου USB 3.0 υψηλού εύρους ζώνης σε επιλεγμένα chipset USB 3.0 (με συμβατότητα προς τα πίσω με USB 2.0).

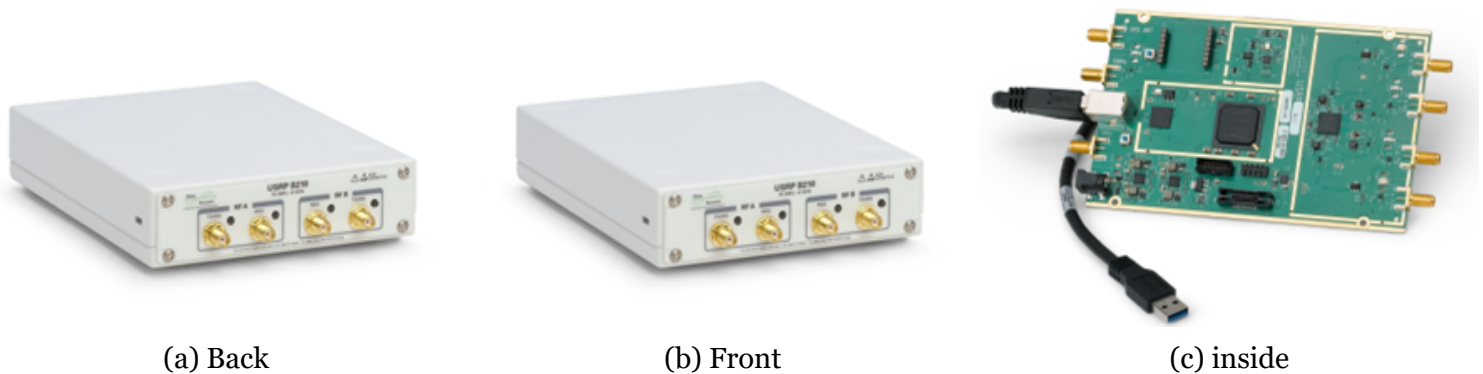


Figure 2: Το FPGA υλισμικό του USRP

I.II GNU Radio Software

Το GNU Radio ^{1 2} είναι μια ελεύθερη και ανοιχτού κώδικα εργαλειοθήκη ανάπτυξης λογισμικού που παρέχει μπλοκ επεξεργασίας σήματος για την υλοποίηση ραδιοφώνων λογισμικού. Μπορεί να χρησιμοποιηθεί με άμεσα διαθέσιμο εξωτερικό υλικό RF χαμηλού κόστους για τη δημιουργία ραδιοφώνων που καθορίζονται από λογισμικό ή χωρίς υλικό σε περιβάλλον προσομοίωσης. Χρησιμοποιείται ευρέως από χομπίστες, σε ακαδημαϊκά και εμπορικά περιβάλλοντα για την υποστήριξη τόσο της έρευνας ασύρματων επικοινωνιών όσο και των πραγματικών ραδιοσυστημάτων. Το GNU Radio εκτελεί όλη την επεξεργασία σήματος. Μπορείτε να το χρησιμοποιήσετε για να γράψετε εφαρμογές, για να λάβετε δεδομένα από ψηφιακές ροές ή για να προωθήσετε δεδομένα σε ψηφιακές ροές, οι οποίες στη συνέχεια μεταδίδονται χρησιμοποιώντας υλικό. Το GNU Radio έχει φίλτρα, κωδικούς καναλιών, στοιχεία συγχρονισμού, ισοσταθμιστές, αποδιαμορφωτές, αποκωδικοποιητές και πολλά άλλα στοιχεία (που ονομάζονται μπλοκ) τα οποία συνήθως βρίσκονται σε ραδιοσυστήματα. Το πιο σημαντικό, περιλαμβάνει μια μέθοδο σύνδεσης αυτών των μπλοκ και στη συνέχεια διαχειρίζεται τον τρόπο με τον οποίο τα δεδομένα μεταφέρονται από το ένα μπλοκ στο άλλο. Η επέκταση του GNU Radio είναι επίσης αρκετά εύκολη. Εάν βρείτε ένα συγκεκριμένο μπλοκ που λείπει, μπορείτε γρήγορα να το δημιουργήσετε και να το προσθέσετε γράφοντας κώδικα Python. Δεδομένου ότι το GNU Radio είναι λογισμικό, μπορεί να χειριστεί μόνο

¹wiki.gnuradio.org

²gnuradio.org

ψηφιακά δεδομένα. Συνήθως, μιγαδικά δείγματα βασικής ζώνης είναι ο τύπος δεδομένων εισόδου για δέκτες και ο τύπος δεδομένων εξόδου για πομπούς. Στη συνέχεια, χρησιμοποιείται αναλογικό υλικό για τη μετατόπιση του σήματος στην επιθυμητή κεντρική συχνότητα. Πέρα από αυτή την απαίτηση, οποιοσδήποτε τύπος δεδομένων μπορεί να περάσει από το ένα μπλοκ στο άλλο - είτε πρόκειται για bit, byte, διανύσματα, ριπές ή πιο σύνθετους τύπους δεδομένων. Οι εφαρμογές GNU Radio γράφονται κυρίως χρησιμοποιώντας τη γλώσσα προγραμματισμού Python, ενώ η παρεχόμενη, κρίσιμη για την απόδοση διαδρομή επεξεργασίας σήματος υλοποιείται σε C++ χρησιμοποιώντας επεκτάσεις κινητής υποδιαστολής επεξεργαστή, όπου είναι διαθέσιμες. Έτσι, ο προγραμματιστής είναι σε θέση να υλοποιήσει ραδιοσυστήματα υψηλής απόδοσης σε πραγματικό χρόνο σε ένα απλό στη χρήση, γρήγορο περιβάλλον ανάπτυξης εφαρμογών. Το GNU Radio είναι ένα πλαίσιο που επιτρέπει στους χρήστες να σχεδιάζουν, να προσομοιώνουν και να αναπτύσσουν εξαιρετικά ικανά ραδιοσυστήματα πραγματικού κόσμου. Πρόκειται για ένα εξαιρετικά αρθρωτό πλαίσιο προσανατολισμένο στο "διάγραμμα ροής" που συνοδεύεται από μια ολοκληρωμένη βιβλιοθήκη μπλοκ επεξεργασίας που μπορούν εύκολα να συνδυαστούν για να κάνουν πολύπλοκες εφαρμογές επεξεργασίας σήματος.



II I/Q Data

Τα δεδομένα in-phase/quadrature (I/Q) είναι μια σύνθετη αναπαράσταση του μεταδιδόμενου σήματος, για το οποίο μπορούν όχι μόνο να δώσουν πληροφορίες σχετικά με το πλάτος των σημάτων αλλά και τις πληροφορίες φάσης. Τα δεδομένα IQ μπορούν να αναπαρασταθούν ως δύο κανάλια δεδομένων που είναι ορθογώνια μεταξύ τους, όπως φαίνεται στο σχήμα (3).

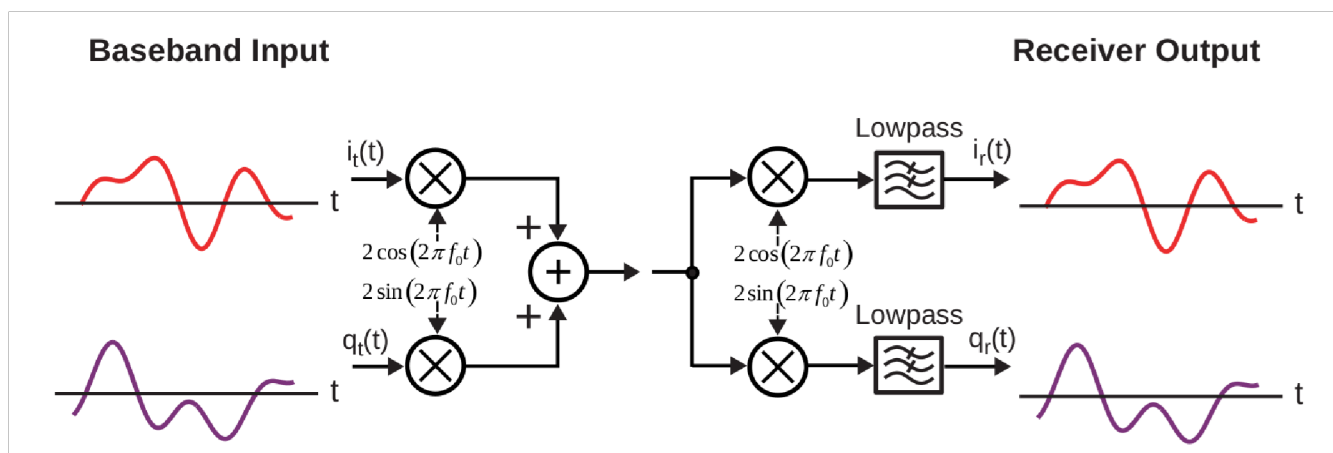


Figure 3: Κανάλια In-Phase/Quadrature

Τα δύο κανάλια στην πλευρά του πομπού μπορούν να γραφτούν ως εξής:

$$\begin{cases} i(t) = i_t(t) \cos(2\pi f_0 t) \\ q(t) = q_t(t) \sin(2\pi f_0 t) \end{cases} \quad (1)$$

Και προσθέτοντας κατά μέλη:

$$y(t) = i_t(t) \cos(2\pi f_0 t) + q_t(t) \sin(2\pi f_0 t) \quad (2)$$

Από την μεριά του δέκτη τα δύο κανάλια μπορούν να ανακτηθούν χρησιμοποιώντας την τριγωνομετρική ταυτότητα γινομένου \rightarrow άθροισμα.³

$$\begin{aligned} i_r(t) &= 2[i_t(t) \cos(2\pi f_0 t) + q_t(t) \sin(2\pi f_0 t)] \sin(2\pi f_0 t) \\ &= [i_t(t) + i_t(t) \cos(4\pi f_0 t)] + [q_t(t) \sin(4\pi f_0 t) + q_t(t) \sin(0)] \\ &= [i_t(t) + i_t(t) \cos(4\pi f_0 t)] + [q_t(t) \sin(4\pi f_0 t)] \end{aligned} \quad (3)$$

$$\begin{aligned} q_r(t) &= 2[i_t(t) \cos(2\pi f_0 t) + q_t(t) \sin(2\pi f_0 t)] \sin(2\pi f_0 t) \\ &= [i_t(t) \sin(4\pi f_0 t) - i_t(t) \sin(0)] + [q_t(t) - q_t(t) \cos(4\pi f_0 t)] \\ &= [i_t(t) \sin(4\pi f_0 t)] + [q_t(t) - q_t(t) \cos(4\pi f_0 t)] \end{aligned} \quad (4)$$

Χρησιμοποιώντας ένα χαμηλοπερατό φίλτρο στον δέκτη τα κανάλια I&Q μπορούν να ανακτηθούν.

III Διαδικασία Εργαστηρίου

III.I Δημιουργώντας ένα μπλόκ διάγραμμα

Σε αυτήν την άσκηση θα παράξετε ένα συνημίτονο κύμα χρησιμοποιώντας το GNU Radio.

- Ανοίξτε ένα καινούριο διάγραμμα ροής.
- Δύο πλαίσια θα βρίσκονται στο διάγραμμα ροής,
 - "Επιλογές", το οποίο χρησιμοποιείται για τον ορισμό καθολικών παραμέτρων για το διάγραμμα ροής.
 - "Μεταβλητή", το οποίο μπορείτε να του δώσετε ένα όνομα και να το χρησιμοποιήσετε στο διάγραμμα ροής σας (προεπιλεγμένη μεταβλητή είναι ο ρυθμός δειγματοληψίας⁴)
- Ορίστε τις επιλογές δημιουργίας σε QT GUI⁵.
- Προσθέστε ένα άλλο μπλοκ **μεταβλητής**, ονομάστε το **freq** και ορίστε το σε **1000**.
- Προσθέστε μια **πηγή σήματος** και ορίστε τον **τύπο εξόδου** σε **float**, **κυματομορφή** σε **συνημίτονο**, **συχνότητα** σε **freq**, **πλάτος** σε **1** και τέλος **μετατόπιση** σε **0**⁶.
- Προσθέστε μια **Throttle** και ορίστε το **Type** σε **Float**⁷.
- Προσθέστε ένα **QT GUI Sink** και ορίστε το **Type** σε **Float** και το **FFT Size** σε **1024**⁸.
- Αλλάξτε τη συχνότητα της πηγής σήματος από 1k σε 2.5k, 5k, 16k και 100k και παρατηρήστε το πεδίο χρόνου και συχνότητας;
- Ερώτηση 1: Ποια είναι η διαφορά και γιατί;

Τελικά, θα πρέπει να έχετε ένα διάγραμμα ροής όπως φαίνεται στο σχήμα (4).

Και όταν εκτελέσετε το διάγραμμα ροής θα λάβετε ένα αποτέλεσμα όπως φαίνεται στο σχήμα (5).

³ $\cos(x) \times \cos(y) = \frac{1}{2}(\cos(x-y) + \cos(x+y))$, $\sin(x) \times \sin(y) = \frac{1}{2}(\cos(x-y) - \cos(x+y))$
 $\sin(x) \times \cos(y) = \frac{1}{2}(\sin(x+y) + \sin(x-y))$, $\cos(x) \times \sin(y) = \frac{1}{2}(\sin(x+y) - \sin(x-y))$

⁴Σκεφτείτε ότι ο ρυθμός δειγματοληψίας χρησιμοποιείται για τον υπολογισμό του διακριτού μεγέθους βημάτων από το ένα δείγμα στο επόμενο μέσα σε μια λειτουργία DSP. Ο ρυθμός δειγματοληψίας αναφέρεται επίσης στο ρυθμό με τον οποίο τα δείγματα διέρχονται από το διάγραμμα ροής. Εάν δεν υπάρχει έλεγχος ρυθμού, ρολόι υλικού ή μηχανισμός στραγγαλισμού, τα δείγματα θα παραχθούν, θα περάσουν από το διάγραμμα ροής και θα καταναλωθούν όσο το δυνατόν γρηγορότερα (δηλαδή το διάγραμμα ροής θα είναι δεσμευμένο από την CPU). Μόνο ένα μπλοκ που αντιπροσωπεύει κάποιο υποκείμενο υλικό με το δικό του ρολόι (π.χ. USRP, κάρτα ήχου) ή το throttle μπλοκ, θα χρησιμοποιήσει το "Sample Rate" για να ρυθμίσει αυτό το ρολόι υλικού και επομένως θα έχει ως αποτέλεσμα την εφαρμογή ελέγχου ρυθμού στα δείγματα στο διάγραμμα ροής.

⁵Προκειμένου να οριστεί το γραφικό περιβάλλον

⁶Ετσι δημιουργείται ένα συνημίτονο σήμα χωρίς αρχική φάση, εκφρασμένο ως ένα διάνυσμα αριθμών κινητής υποδιαστολής.

⁷Προκειμένου ο μέσος ρυθμός δεδομένων να μην υπερβεί τον ρυθμό δειγματοληψίας σε sps(σύμβολα/δευτερόλεπτο)

⁸Ωστε τα λαμβανόμενα δεδομένα να παρασταθούν στην οθόνη.

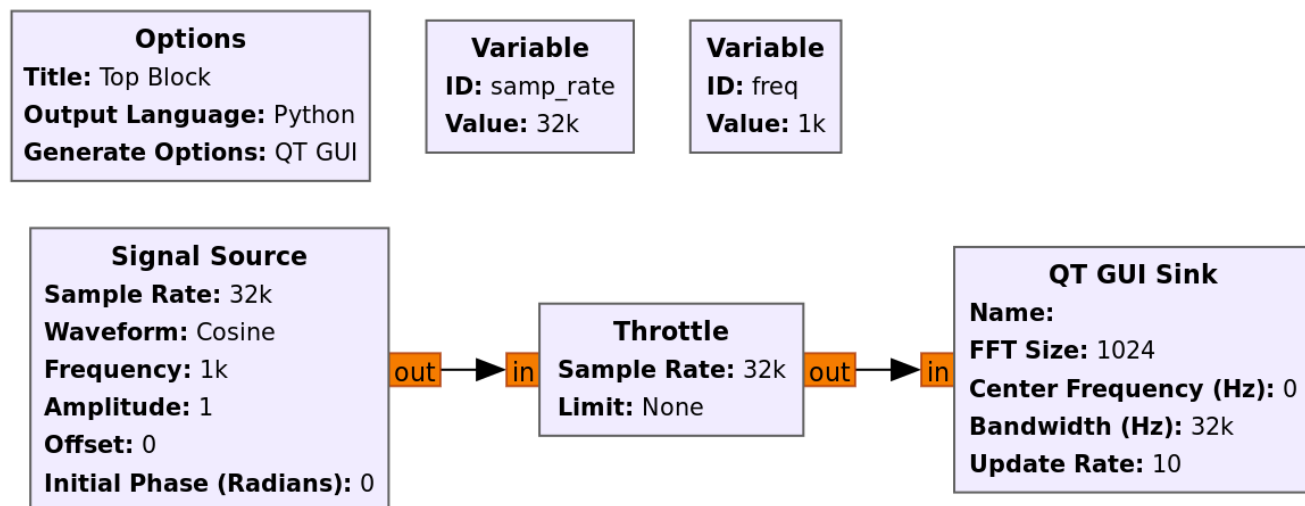


Figure 4: Πηγή Σήματος GNU Radio

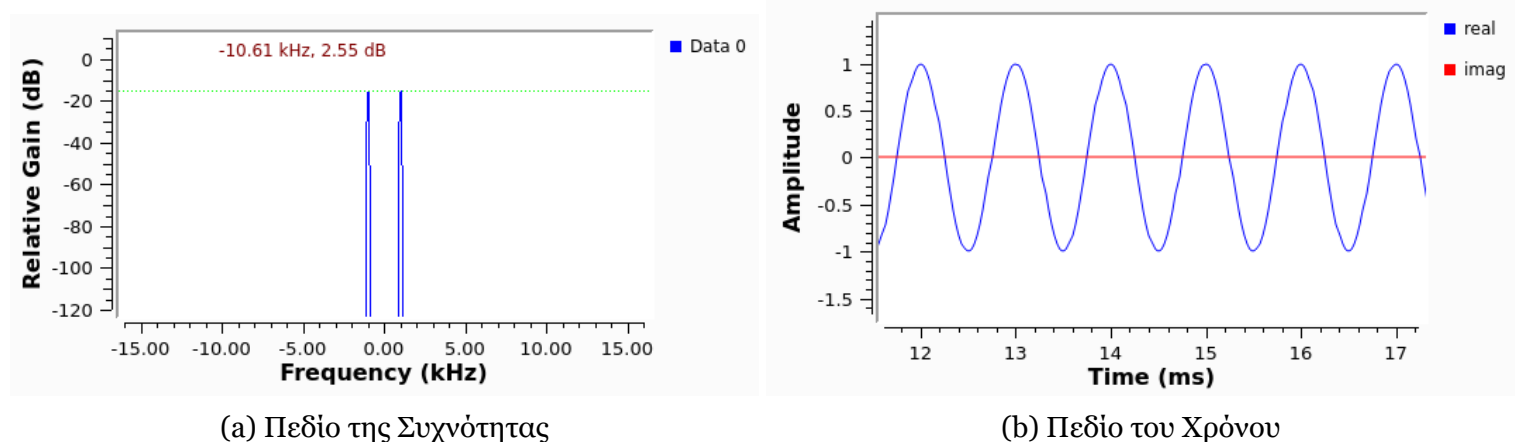


Figure 5: Έξοδος του διαγράμματος ροής

III.II Μιγαδικό Σήμα με Θόρυβο

Σε αυτή την άσκηση, θα δημιουργήσετε ένα μιγαδικό κύμα συνημιτόνου με θόρυβο.

- Ανοίξτε ένα καινούριο διάγραμμα ροής
- Ορίστε τις **επιλογές δημιουργίας** σε **QT GUI**.
- Ορίστε τον **ρυθμό δειγματοληψίας** σε **64000**.
- Προσθέστε ένα άλλο μπλοκ **μεταβλητής**, ονομάστε το **freq** και ορίστε το σε **2000**.
- Προσθέστε μια **πηγή σήματος** και ορίστε τον **τύπο εξόδου** σε **complex**, **κυματομορφή** σε **συνημίτονο**, **συχνότητα** σε **freq**, **πλάτος** σε **1** και τέλος **μετατόπιση** σε **0**.
- Προσθέστε μια **πηγή θορύβου** και ορίστε τον **τύπο εξόδου** σε **μιγαδικούς αριθμούς**, τον **Τύπο θορύβου** σε **Gaussian** και το **Πλάτος** σε **0.1**.
- Προσθέστε ένα μπλόκ **πρόσθεσης**.
- Προσθέστε ένα **Throttle** και ορίστε το **Type** σε **Complex**.
- Προσθέστε ένα **QT GUI Sink** και ορίστε το **Type** σε **Complex** και το **FFT Size** σε **1024**.
- Ερώτηση 2: Πόσες κορυφές συχνότητας μπορείτε να δείτε και γιατί;
- Ερώτηση 3: Πώς μπορείτε να μετριάσετε το φαινόμενο του θορύβου;

Τελικά, θα πρέπει να έχετε ένα διάγραμμα ροής όπως φαίνεται στο σχήμα (6).

Και όταν εκτελέσετε το διάγραμμα ροής θα λάβετε ένα αποτέλεσμα όπως φαίνεται στο σχήμα (7).

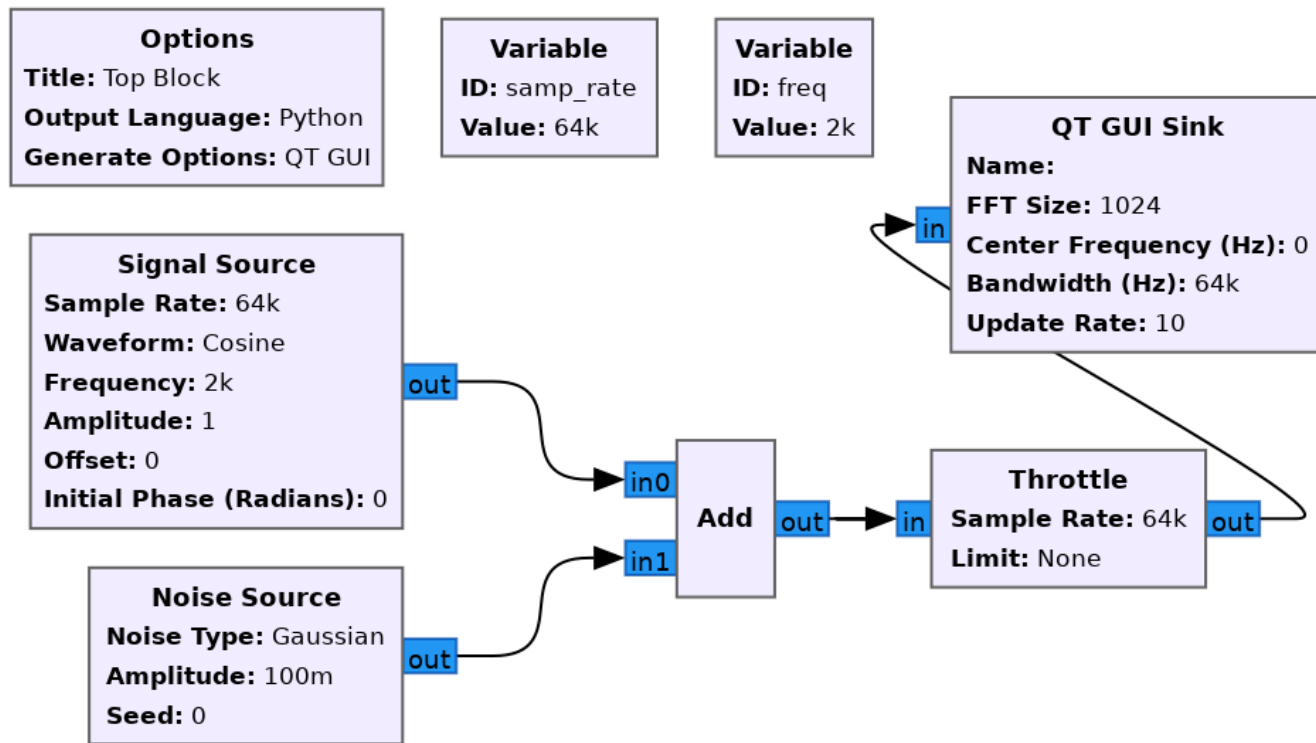
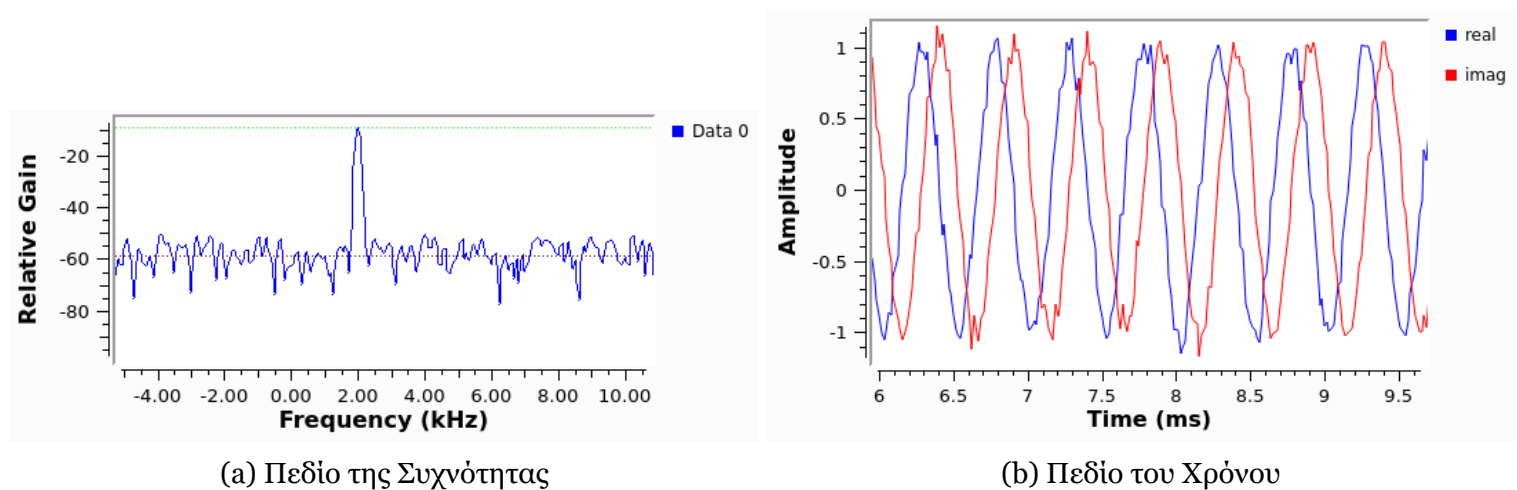


Figure 6: Σύνθετο σήμα με θόρυβο



(a) Πεδίο της Συχνότητας

(b) Πεδίο του Χρόνου

Figure 7: Έξοδος του διαγράμματος ροής

III.III Συνδέοντας το USRP

Σε αυτή την άσκηση θα μεταδώσετε και θα λάβετε ένα σύνθετο κύμα συνημιτόνου χρησιμοποιώντας το USRP.

- Ανοίξτε ένα καινούριο διάγραμμα ροής
- Ορίστε τις **επιλογές δημιουργίας** σε QT GUI.
- Ορίστε τον **ρυθμό δειγματοληψίας** σε **1e6**.
- Προσθέστε ένα άλλο μπλοκ **μεταβλητής**, ονομάστε το **freq** και ορίστε το σε **20e3**.
- Προσθέστε ένα άλλο μπλοκ **μεταβλητής**, ονομάστε το **carrier_freq** και ορίστε το σε **1e9**.
- Προσθέστε μια **πηγή σήματος** και ορίστε τον **τύπο εξόδου** σε **complex**, **κυματομορφή** σε **συνημίτονο**, **συχνότητα** σε **freq**, **πλάτος** σε **1** και τέλος **μετατόπιση** σε **0**.
- Προσθέστε ένα **UHD: USRP Sink** και ορίστε τον **τύπο εισόδου** σε **complex float32**, την **Κεντρική Συχνότητα** σε **carrier_freq** και την **Τιμή Απολαβής** σε **1**.
- Προσθέστε ένα **UHD: USRP Source** και ορίστε τον **τύπο εξόδου** σε **complex float32**, την **Κεντρική Συχνότητα** σε **carrier_freq** και την **Τιμή Απολαβής** σε **0**.

- Προσθέστε ένα **QT GUI Sink** και ορίστε το **Type** σε **Complex** και το **FFT Size** σε **1024**.
- Προσθέστε ένα φίλτρο ώστε να ανακτήσετε το σωστό σήμα με λιγότερο θόρυβο.
- Ερώτηση 4: Ποιο φίλτρο χρησιμοποιείτε και ποιες είναι οι παράμετροι που θα επιλέξετε;

Στο τέλος θα πρέπει να έχετε ένα διάγραμμα ροής όπως φαίνεται στην εικόνα (8).

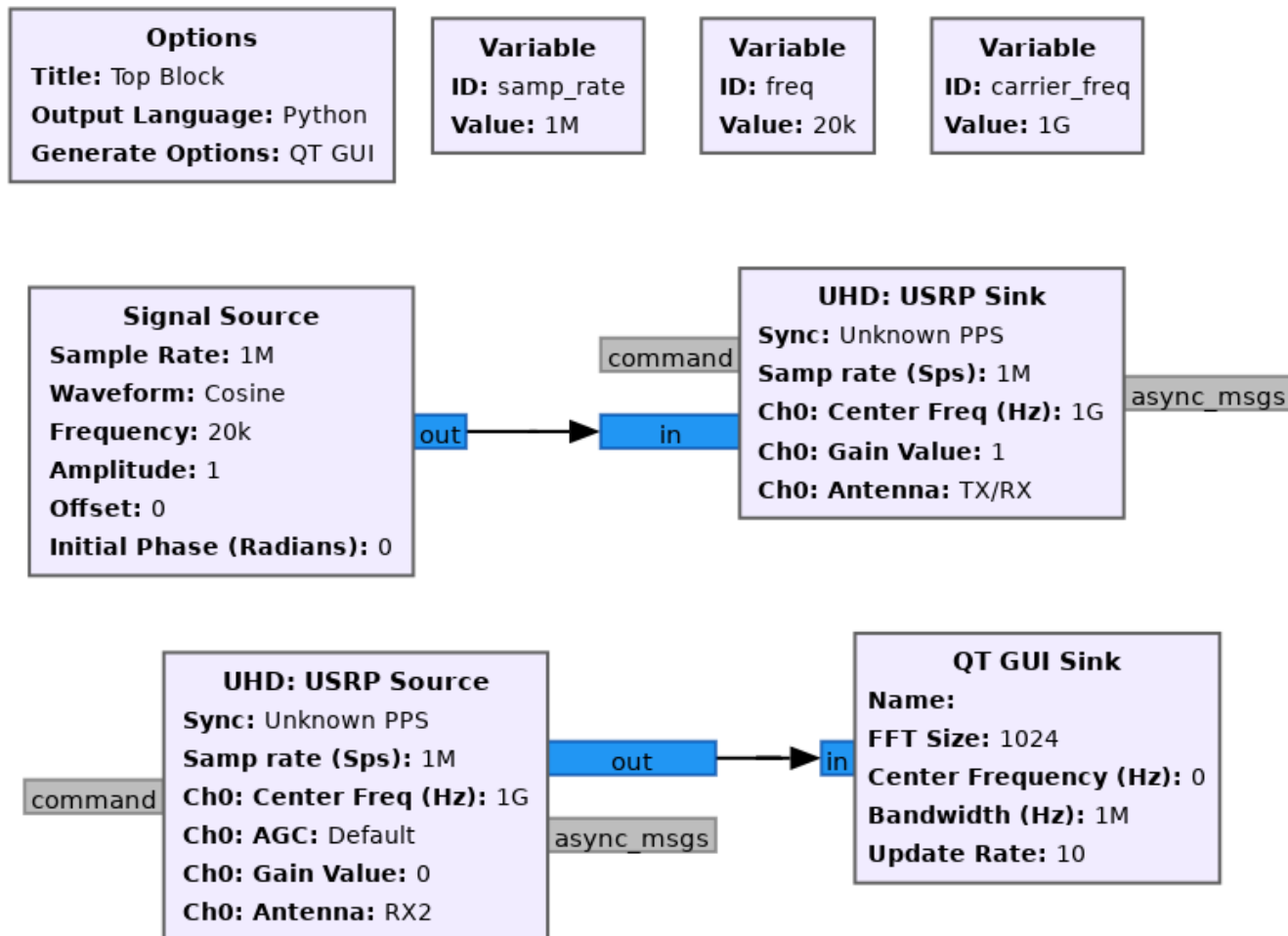


Figure 8: USRP ως Πηγή Σήματος