

**1ο ΕΠΑΛ- Ε.Κ. Συκεών -Τομέας:** Ηλεκτρονικής, Ηλεκτρολογίας και Αυτοματισμού

**Εκπαιδευτικοί:** Μπουλταδάκης Στέλιος – Μαυρίδης Κώστας

**Μάθημα:** Ρομποτική

**Αντικείμενο :** Μεταφορά δεδομένων μέσω του πρωτοκόλλου RS232 μεταξύ Arduino UNO R3 και Η/Υ με εφαρμογή -VI LabView

**ΒΙΒΛΙΟΓΡΑΦΙΑ:**

Το φύλλο έργου στηρίχτηκε σε πληροφορίες

1. από την ηλεκτρονική διεύθυνση: <https://www.arduino.cc>
2. σε άλλα φύλλα έργου του σχολείου

**Στόχοι:**

α) κατανόηση της λειτουργίας του σειριακού πρωτοκόλλου RS232 τόσο από την πλευρά της πλακέτας Arduino UNO R3 καθώς και από την πλευρά μιας εφαρμογής εικονικού οργάνου σε LabView

**Τα Υλικά που θα χρειαστούμε:**

1. Πλακέτα Arduino UNO,
2. Καλώδια σύνδεσης.

**Βήμα 1ο:** (Εγκατάσταση Οδηγού πλακέτας Arduino για την πρώτη φορά αλλιώς αγνοούμε το Βήμα 1ο).

Συνδέουμε το ARDUINO. σε μια θύρα USB του Η/Υ (αν δεν έχει εγκατασταθεί ο driver ) αφήνουμε το λειτουργικό σύστημα να κάνει αναζήτηση οδηγού στο διαδίκτυο, κατέβασμα του οδηγού στον Η/Υ και εγκατάσταση.

**Βήμα 2ο:** Εύρεση της θύρας στην οποία το λειτουργικό σύστημα βλέπει την πλακέτα Arduino.

A) Κάνουμε δεξί κλικ στο Η/Υ (windows 10: δεξί κλικ στο εικονίδιο WINDOWS κάτω αριστερά στην οθόνη) και επιλέγουμε Διαχείριση Συσκευών/Θύρες COM & LPT για να δούμε σε ποια θύρα COM (θύρα USB) έχει εγκατασταθεί – συνδεθεί η πλακέτα ARDUINO.

B) Αντίστοιχα δηλώνουμε την ίδια σειριακή θύρα από την οποία θα μεταφερθεί το πρόγραμμα από το IDE Arduino του Η/Υ προς την πλακέτα Arduino.

**Βήμα 3ο:** Στο περιβάλλον IDE Arduino γράφουμε τον παρακάτω κώδικα για τον οποίο ενημερωνόμαστε από τον καθηγητή.

```
int incomingByte = 0; // for incoming serial data
void setup() {
    Serial.begin(9600); // opens serial port, sets data rate to 9600 bps
}
void loop() {
    // send data only when you receive data:
```

```

        if (Serial.available() > 0) {
// read the incoming byte:
        incomingByte = Serial.read();
// say what you got:
        Serial.println(incomingByte);
    }
}

```

**Βήμα 4ο:** Κάνουμε μεταγλώττιση, έλεγχο και μεταφορά του κώδικα σύμφωνα με τις οδηγίες του καθηγητή και απεικονίζουμε τις μετρήσεις θερμοκρασίας που στέλνει το Arduino στο περιβάλλον της γλώσσας προγραμματισμού όπως φαίνεται παρακάτω.

Για τη μεταφορά δεδομένων από το Arduino στην σειριακή οθόνη ή plotter του IDE Arduino επιβεβαιώνουμε ότι έχουμε δηλώσει την σειριακή θύρα-καταχωρητές που αντιστοιχεί στην θύρα USB στην οποία έχουμε συνδέσει το Arduino (Εργαλεία/Σειριακή Οθόνη ή plotter/π.χ. COM5 ) όπως είδαμε από το Βήμα 3.

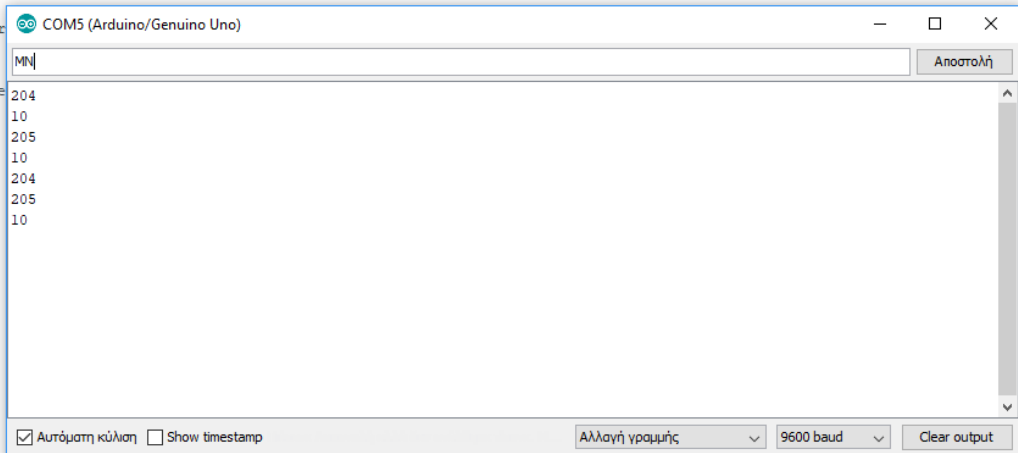
```

int incomingByte = 0; // for incoming serial data

void setup() {
    Serial.begin(9600); // opens serial port, sets baud rate to 9600 bps
}

void loop() {
    // send data only when you receive data:
    if (Serial.available() > 0) {
        // read the incoming byte:
        incomingByte = Serial.read();
        // say what you got:
        // Serial.print("I received: ");
        // Serial.println(incomingByte, DEC);
        Serial.println(incomingByte);
    }
}

```



Παρατηρούμε ότι για κάθε χαρακτήρα που στέλνουμε στο Arduino, αυτό μας επιστρέφει:

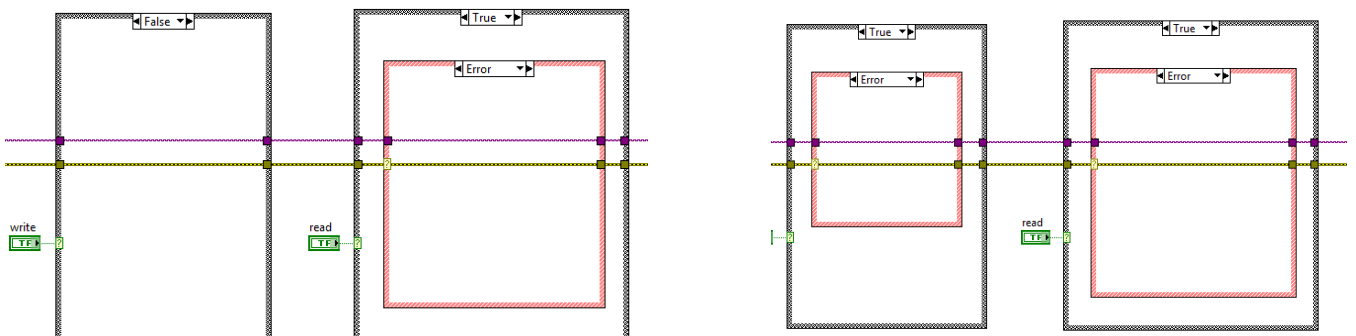
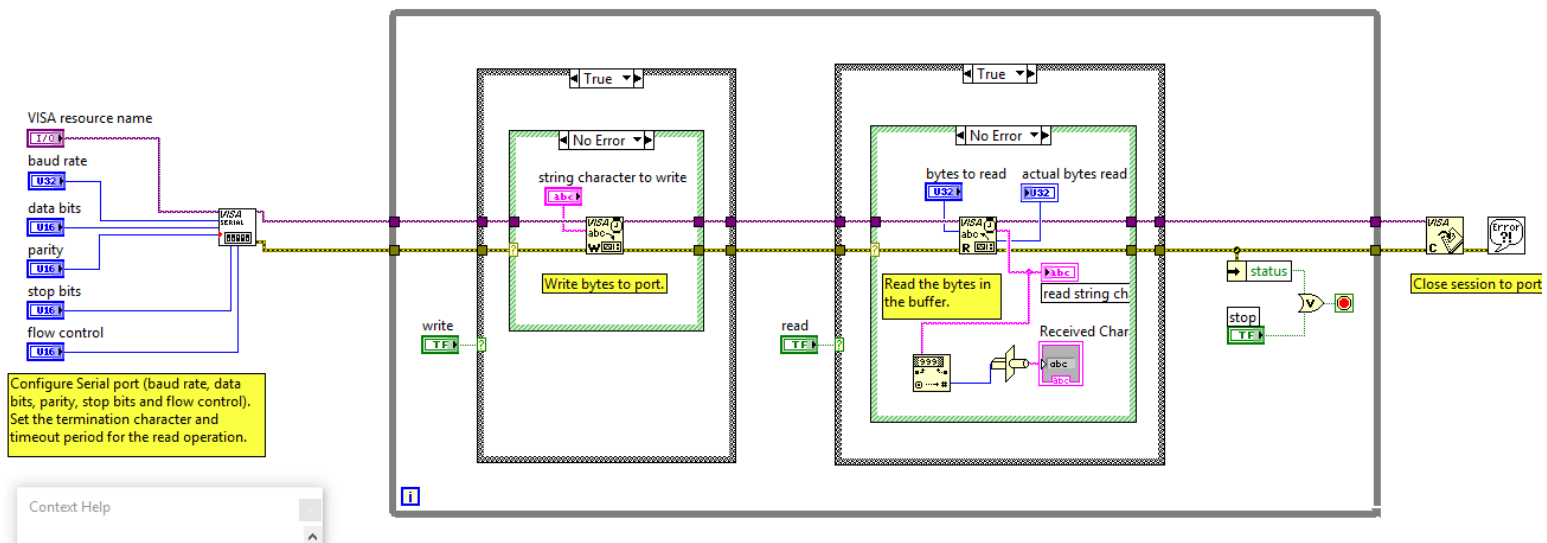
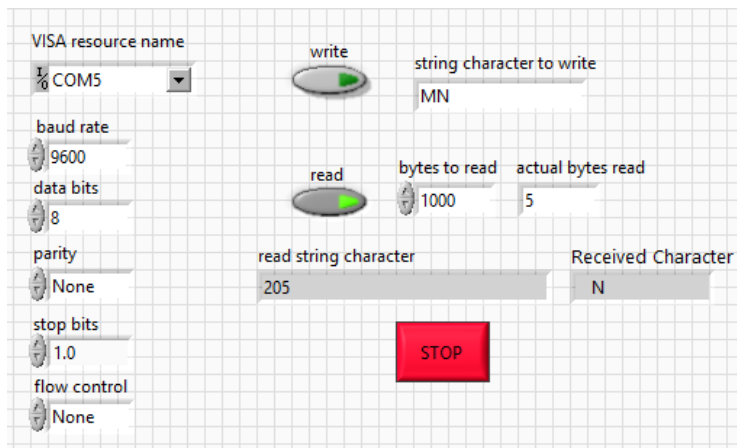
- A) το δεκαδικό ισοδύναμο του χαρακτήρα ASCII Που του στείλαμε
  - B) το δεκαδικό ισοδύναμο (10) του χαρακτήρα LF με το οποίο δηλώνει το τέλος αποστολής χαρακτήρων
- π.χ. για το χαρακτήρα M: επιστρέφει το 204 και το 10  
για το χαρακτήρα N: επιστρέφει το 205 και το 10  
για τους χαρακτήρες MN: επιστρέφει το 205, το 204 και το 10

**Βήμα 6ο:** Στο LabView σχεδιάζουμε το παρακάτω VI.

**ΠΡΟΣΟΧΗ:** Πριν τρέξουμε το VI,

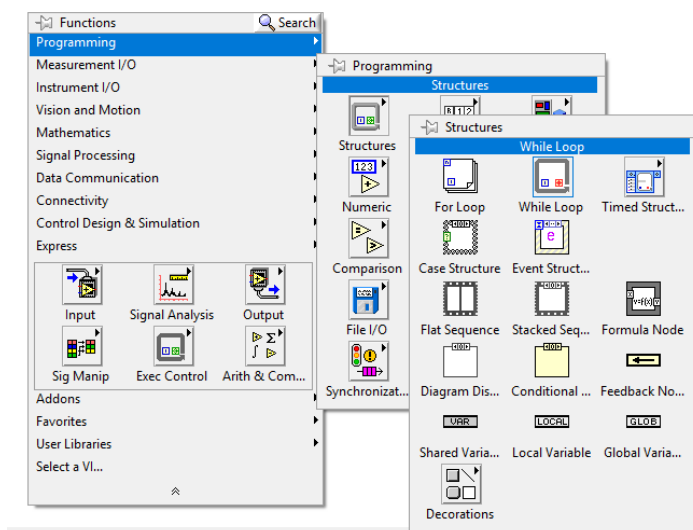
A) απενεργοποιούμε το σειριακό PLOTTER από το IDE Arduino για να μην κάνει conflict με το LabView επειδή χρησιμοποιείται η ίδια USB θύρα του Η/Υ για τη μεταφορά δεδομένων από το Arduino < ---> Σειριακό Plotter του Arduino IDE αλλά και μεταξύ Arduino < ---> VI του LabView.

B) Θέτουμε τη θύρα π.χ. COM5 στο οποίο είναι συνδεδεμένο το LaView

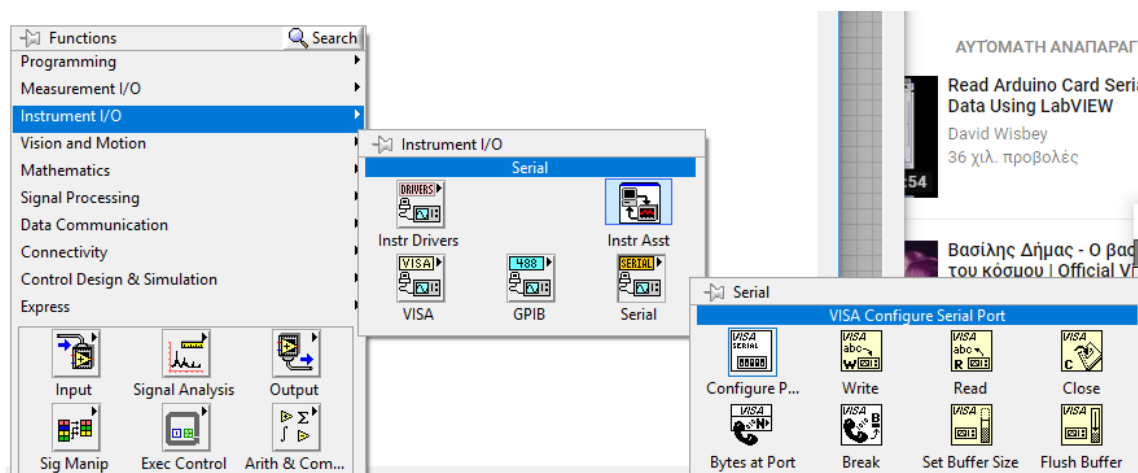


Οι μηχανισμοί που χρησιμοποιούνται παραπάνω βρίσκονται μέσα από τις παρακάτω διαδρομές (ανάλογα αν είμαι στο παράθυρο Front Panel ή στο Block diagram και κάνοντας δεξί κλικ αντίστοιχα).

While..Loop: μηχανισμός για να επαναλαμβάνονται οι μετρήσεις μέχρι να πατήσουμε το πλήκτρο STOP. Η διαδρομή για το μηχανισμό βρίσκεται κάνοντας δεξί κλικ στο παράθυρο Block Diagram /Functions όπως φαίνεται στην παρακάτω εικόνα.



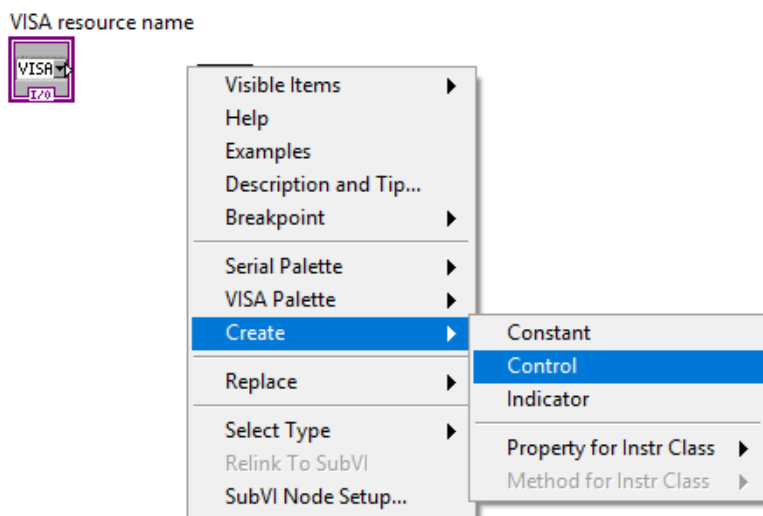
## VISA Configure Serial Port



Οι ιδιότητες του μηχανισμού VISA δίνονται επιλέγοντας μηχανισμούς Control όπως φαίνεται παρακάτω:

VISA Resource name : δεξί κλικ *Create Control* για να ορίζουμε κάθε φορά τη θύρα καταχωρητών COM.. στην οποία συνδέεται το Arduino (είναι οι καταχωρητές τους οποίους βλέπει τη θύρα USB και επομένως την κάρτα Arduino το λειτουργικό σύστημα). Π.χ. COM5. ΠΡΟΣΟΧΗ: Όλοι οι μηχανισμοί VISA πρέπει να έχουν το ίδιο όνομα από τη θύρα που γίνεται η συγκεκριμένη μέτρηση. Αν υπάρχουν δύο διαφορετικές σειριακές

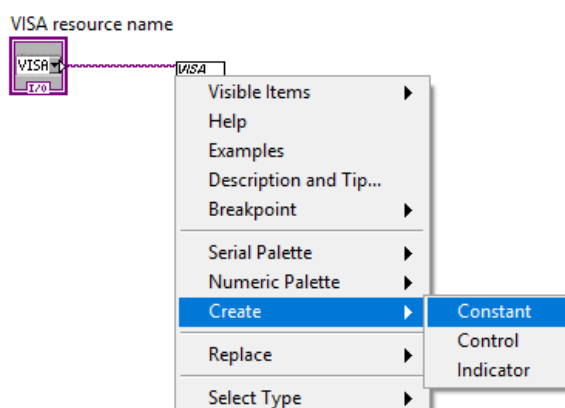
συνδέσεις στον Η/Υ θα πρέπει να έχουμε δύο διαφορετικές ομάδες μηχανισμών VISA μια για κάθε σύνδεση. Το κοινό όνομα αποκαθίσταται με τη σύνδεση του μωβ καλωδίου στο block diagram.



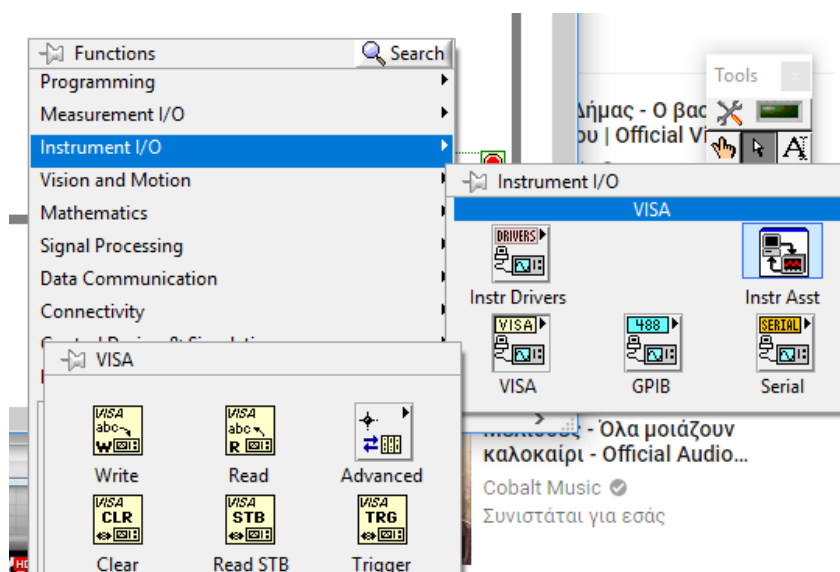
Δηλώνουμε επίσης και τις ιδιότητες της σειριακής επικοινωνίας που αφορούν το Baud rate και τον αριθμό των bits στους οποίους κωδικοποιούνται οι χαρακτήρες δεδομένων που μεταφέρονται μέσω της σειριακής επικοινωνίας. Αυτές οι ιδιότητες πρέπει να είναι ίδιες τόσο για τη σειριακή επικοινωνία από τη μεριά του Arduino όσο και από τη μεριά του Η/Υ. Επειδή όμως οι ιδιότητες είναι καθορισμένες χρησιμοποιούμε τους μηχανισμούς Constant όπως δείχνεται παρακάτω:

VISA Baud Rate: Create Constant → 9600bps

VISA Data Bits: Create Constant → 8 bits

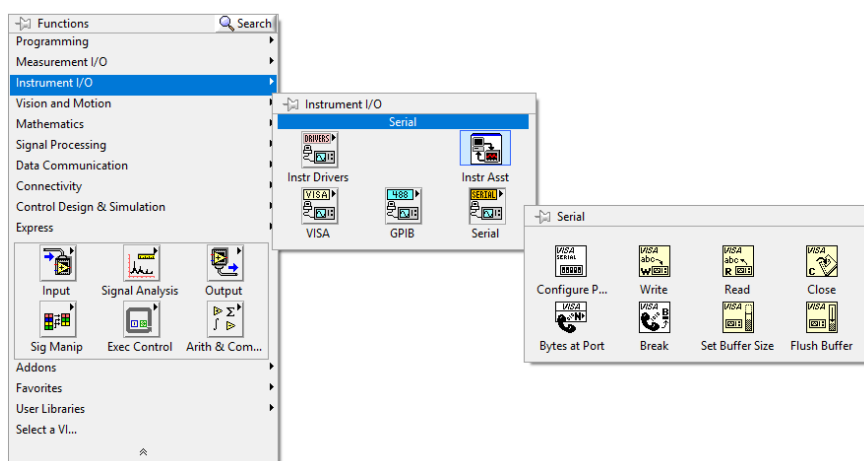


VISA Read : μηχανισμός που διαβάζει ( Read data serial) που προωθούνται από τον προηγούμενο μηχανισμό.



Συνδέουμε και το μηχανισμό Indicator για να φαίνονται οι χαρακτήρες που έρχονται από τη σειριακή επικοινωνία. Η σύνδεση τέτοιων μηχανισμών είναι η γνωστή διαδικασία του LabView όπως φαίνεται παρακάτω.

VISA Close: μηχανισμός που ολοκληρώνει τη σειριακή σύνδεση του εικονικού οργάνου (VI) του LabView με το Arduino μέσω σειριακής επικοινωνίας και με το καλώδιο USB.



Error Handling: μηχανισμός που εμφανίζει μηνύματα ή οδηγίες σε περίπτωση σφάλματος. Όλοι οι μηχανισμοί έχουν διασυνδεμένη την ιδιότητα error μεταξύ τους και με αυτό το μηχανισμό.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ