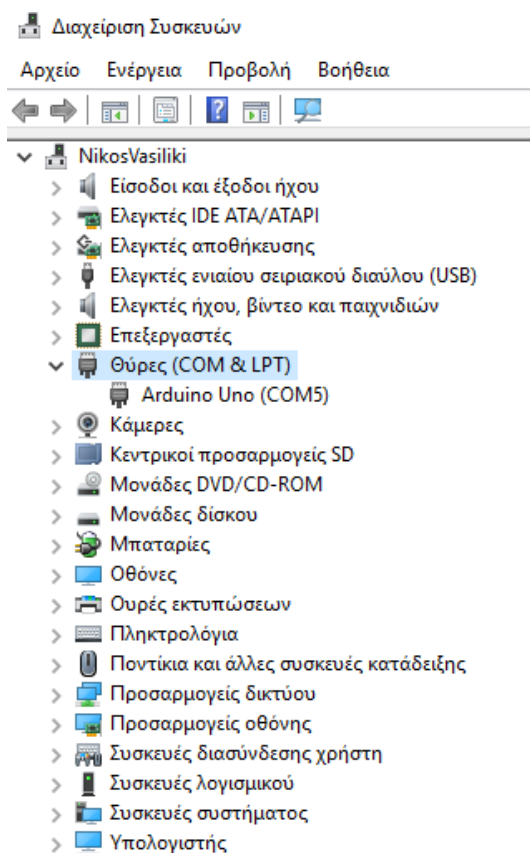
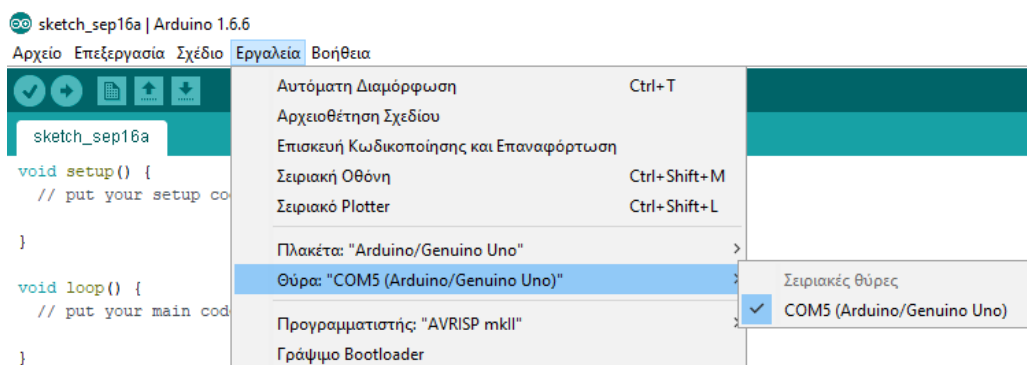


Βήμα 3ο: Εύρεση της θύρας στην οποία το λειτουργικό σύστημα βλέπει την πλακέτα Arduino.

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 4ο: Στο περιβάλλον IDE Arduino γράφουμε τον παρακάτω κώδικα για τον οποίο ενημερωνόμαστε από τον καθηγητή.

```

const int LM35_Pin = A0; // δήλωση του αναλογικού Pin στο οποίο θα συνδεθεί ο αισθητήρας LM35
const float temperature=0.00;

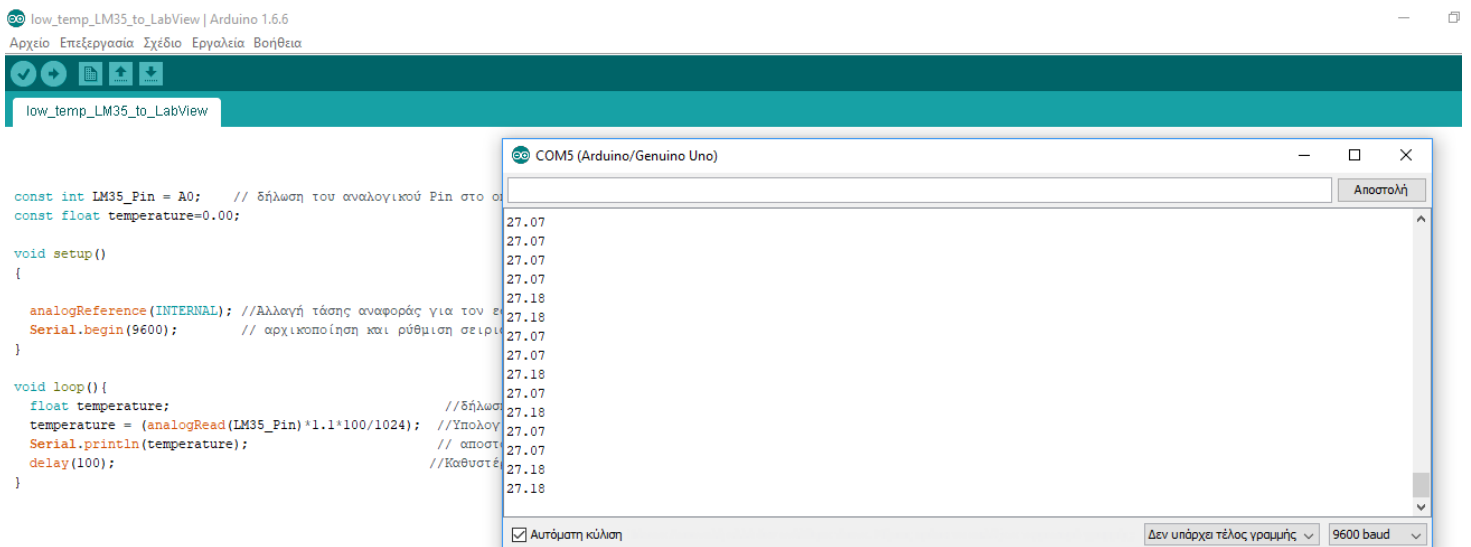
void setup()
{
  analogReference(INTERNAL); //Αλλαγή τάσης αναφοράς για τον εσωτερικό μετατροπέα A/D σε 1.1V
  Serial.begin(9600); // αρχικοποίηση και ρύθμιση σειριακής επικοινωνίας του Arduino με τον Η/Υ
}

void loop(){
  float temperature; //δήλωση της Μεταβλητής για τη θερμοκρασία
  temperature = (analogRead(LM35_Pin)*1.1*100/1024); //Υπολογισμός της θερμοκρασίας
  Serial.println(temperature); // αποστολή της μεταβλητής θερμοκρασίας στο περιφερειακό
σειριακής επικοινωνίας του Arduino
  delay(100); //Καθυστέρηση πριν την επόμενη μέτρηση
}

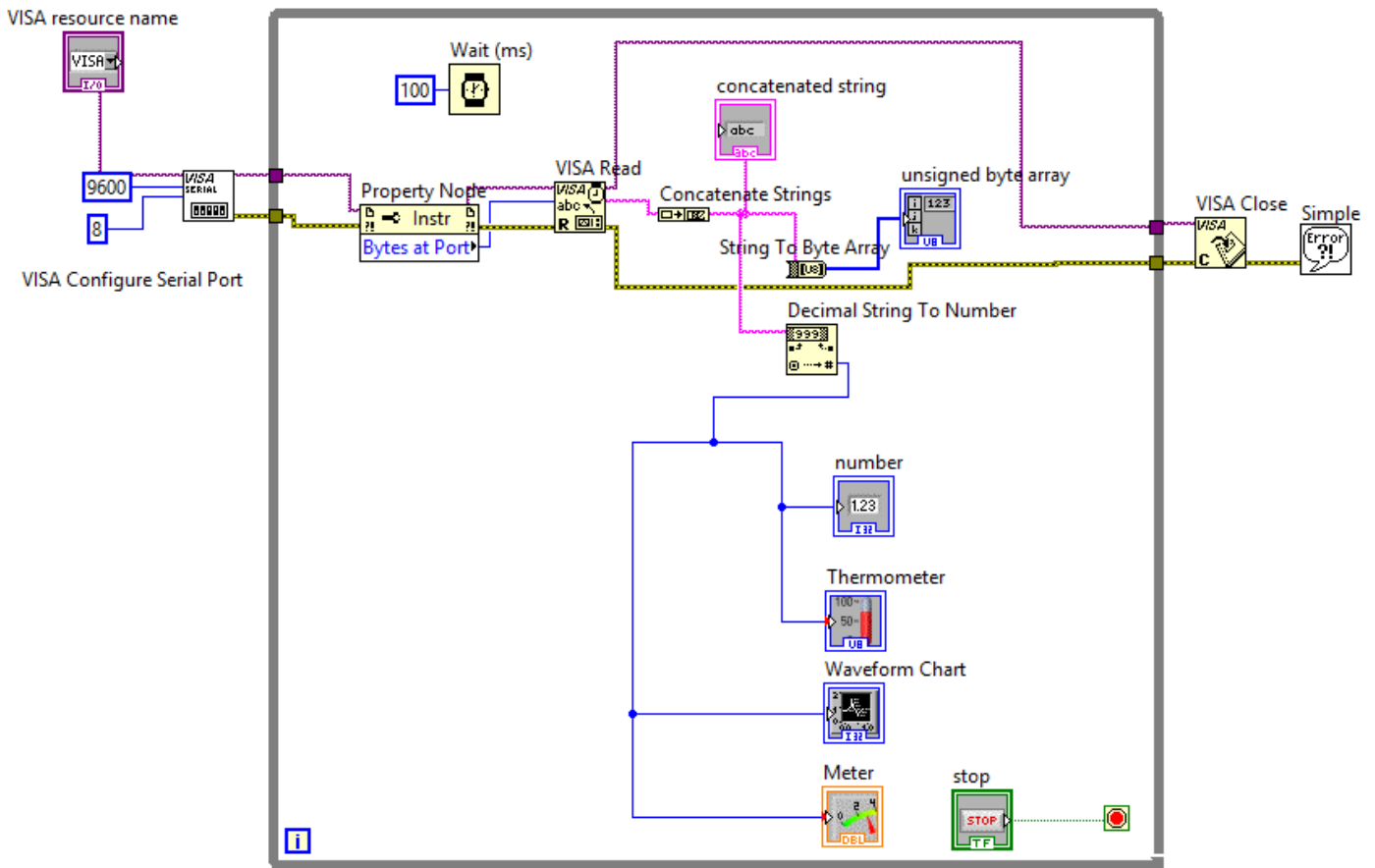
```

Βήμα 5ο: 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.

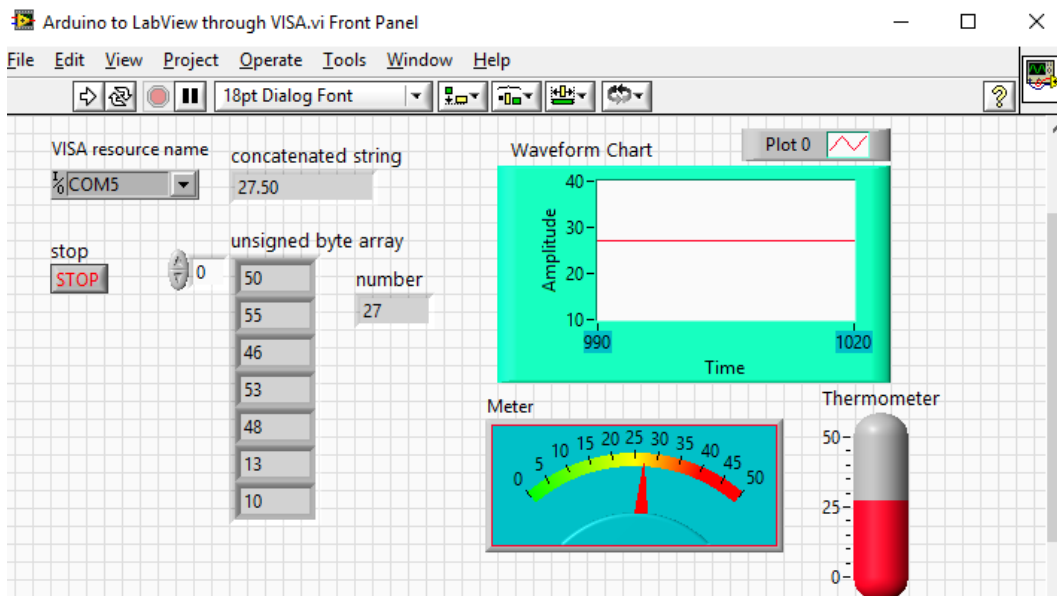
Για τη μεταφορά δεδομένων από το Arduino στην σειριακή οθόνη ή plotter του IDE Arduino επιβεβαιώνουμε ότι έχουμε δηλώσει την σειριακή θύρα-καταχωρητές που αντιστοιχεί στην θύρα USB στην οποία έχουμε συνδέσει το Arduino (Εργαλεία/Σειριακή Οθόνη ή plotter/π.χ. COM5) όπως είδαμε από το Βήμα 3.



Βήμα 6ο: Στο LabView σχεδιάζουμε το παρακάτω VI.

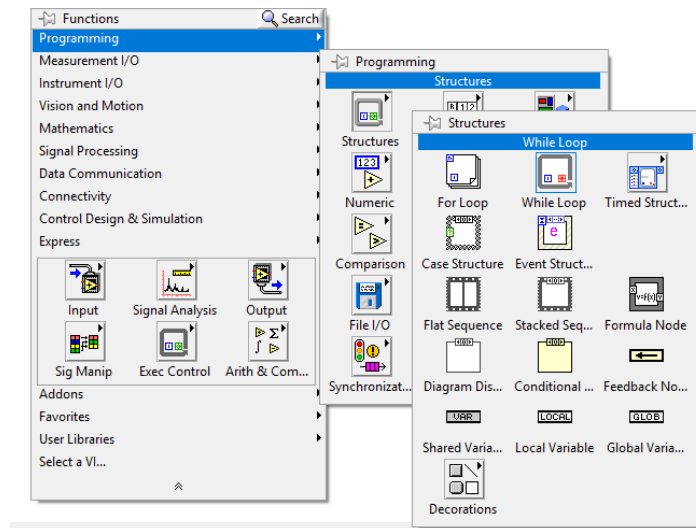


Πριν τρέξουμε το VI, απενεργοποιούμε τη σειριακή οθόνη από το IDE Arduino για να μην κάνει conflict με το LabView επειδή θα χρησιμοποιείται η ίδια USB θύρα για τη μεταφορά δεδομένων από το Arduino στο LabView.

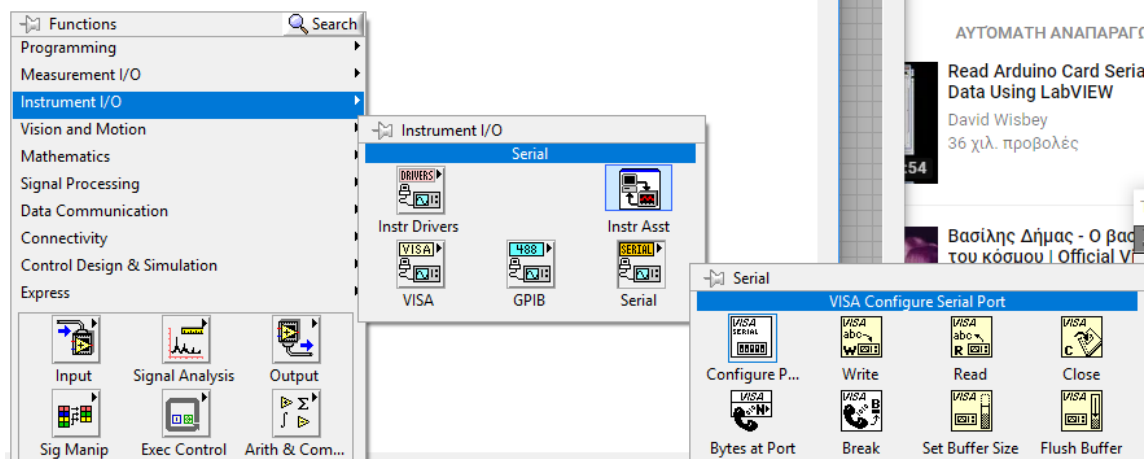


Οι μηχανισμοί που χρησιμοποιούνται παραπάνω βρίσκονται μέσα από τις παρακάτω διαδρομές (ανάλογα αν είμαι στο παράθυρο Front Panel ή στο Block diagram και κάνοντας δεξί κλικ αντίστοιχα).

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

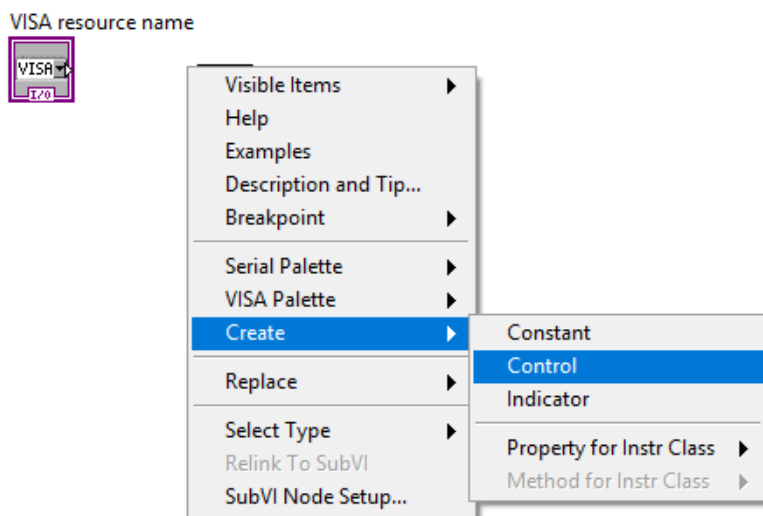


Οι ιδιότητες του μηχανισμού VISA δίνονται επιλέγοντας μηχανισμούς Control όπως φαίνεται παρακάτω:

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

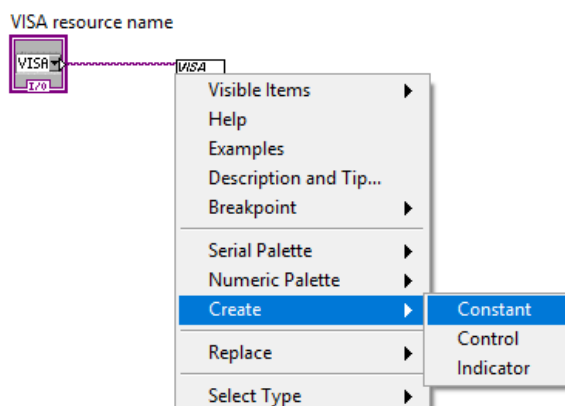
συγκεκριμένη μέτρηση. Αν υπάρχουν δύο διαφορετικές σειριακές συνδέσεις στον Η/Υ θα πρέπει να έχουμε δύο διαφορετικές ομάδες μηχανισμών VISA μια για κάθε σύνδεση. Το κοινό όνομα αποκαθίσταται με τη σύνδεση του μωβ καλωδίου στο *block diagram*.



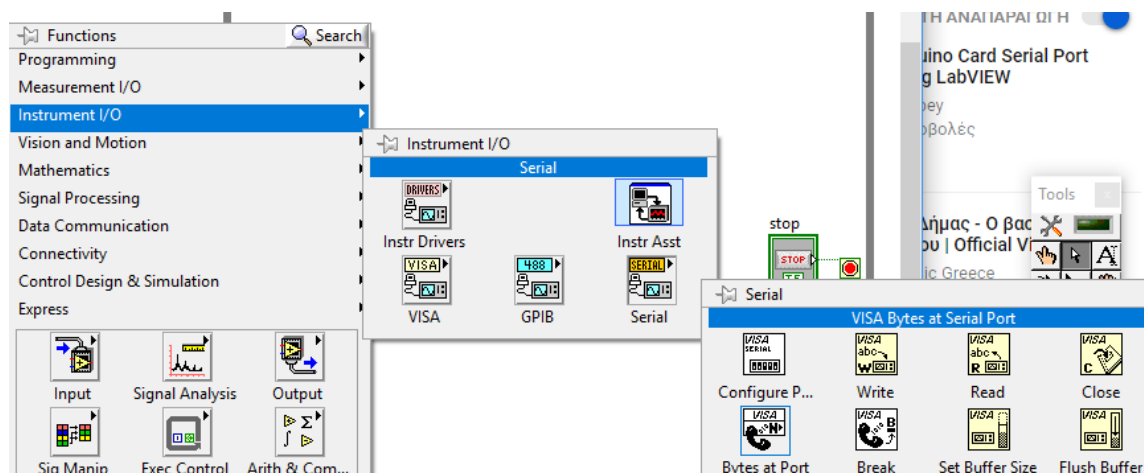
Δηλώνουμε επίσης και τις ιδιότητες της σειριακής επικοινωνίας που αφορούν το Baud rate και τον αριθμό των bits στους οποίους κωδικοποιούνται οι χαρακτήρες δεδομένων που μεταφέρονται μέσω της σειριακής επικοινωνίας. Αυτές οι ιδιότητες πρέπει να είναι ίδιες τόσο για τη σειριακή επικοινωνία από τη μεριά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

VISA Baud Rate: Create Constant → 9600bps

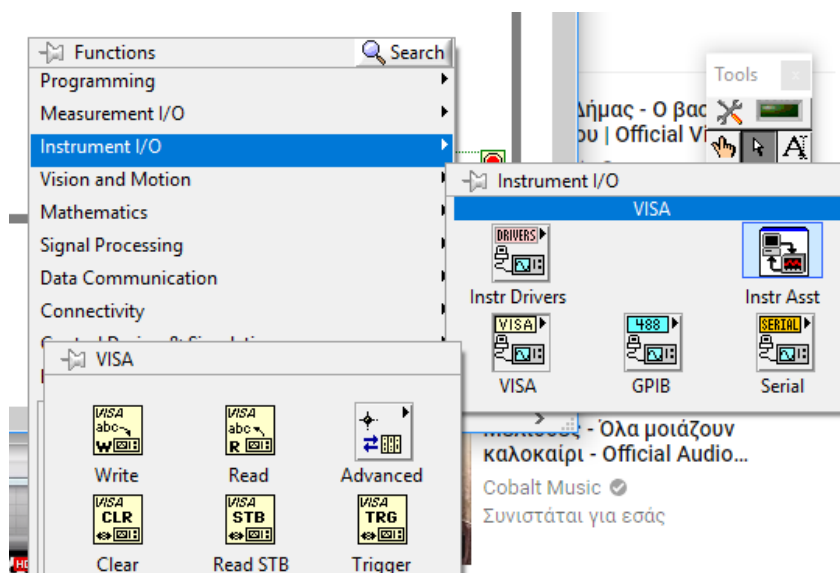
VISA Data Bits: Create Constant → 8 bits



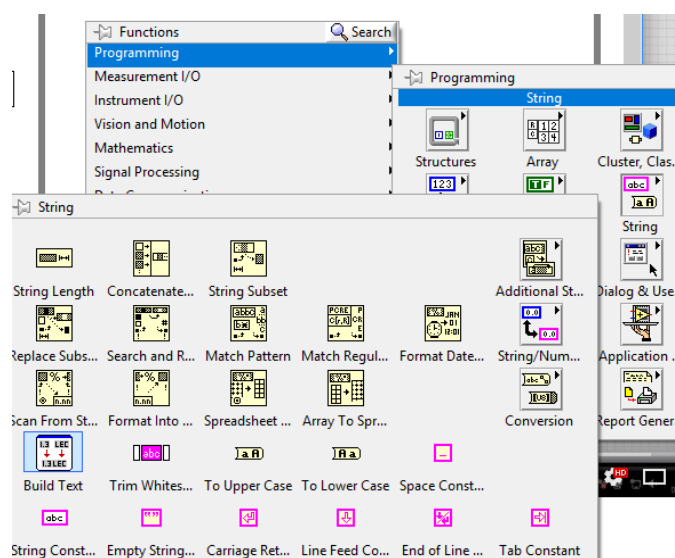
VISA : Bytes at Port είναι ο μηχανισμός που διαβάζει bytes από τη σειριακή –USB θύρα και τα προωθεί στους υπόλοιπους μηχανισμούς του LabView για επεξεργασία.



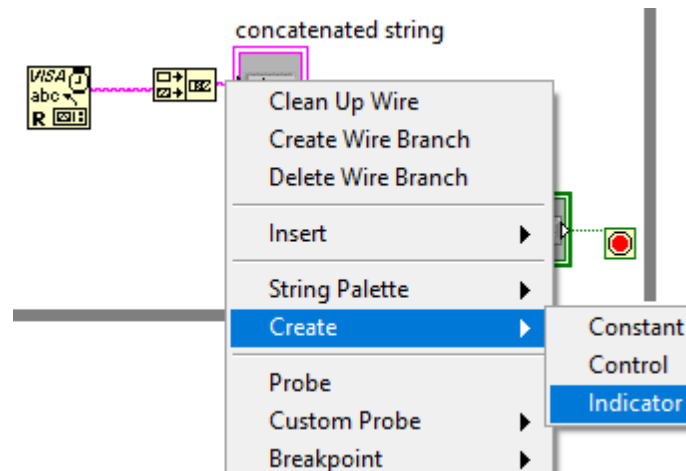
VISA Read : μηχανισμός που διαβάζει (Read data serial) που προωθούνται από τον προηγούμενο μηχανισμό.



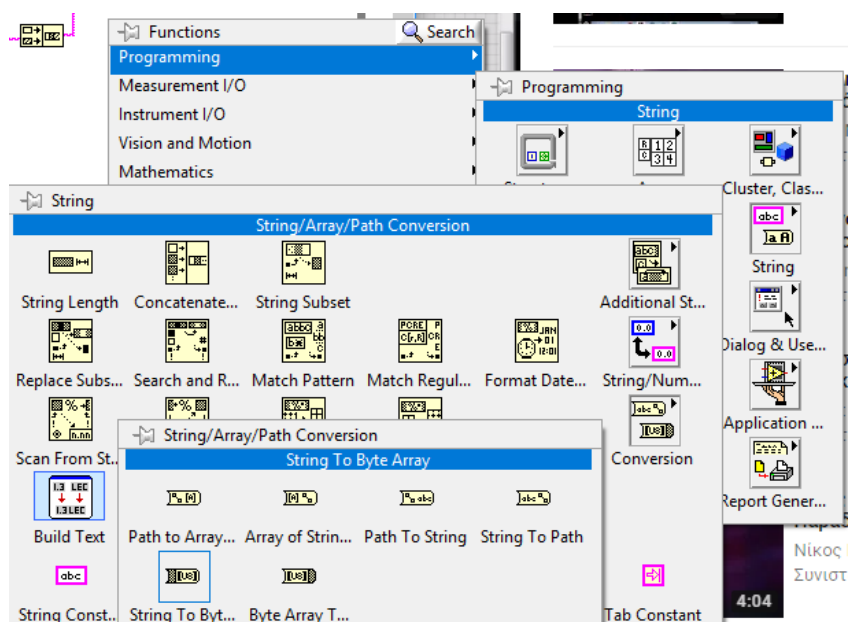
Concatenated String: Μηχανισμός που μαζεύει έναν - έναν τους χαρακτήρες (String) που έρχονται από τη σειριακή επικοινωνία και τους ενοποιεί σε ένα μήνυμα. Η διαδρομή για να βρούμε το μηχανισμό φαίνεται στην παρακάτω εικόνα.



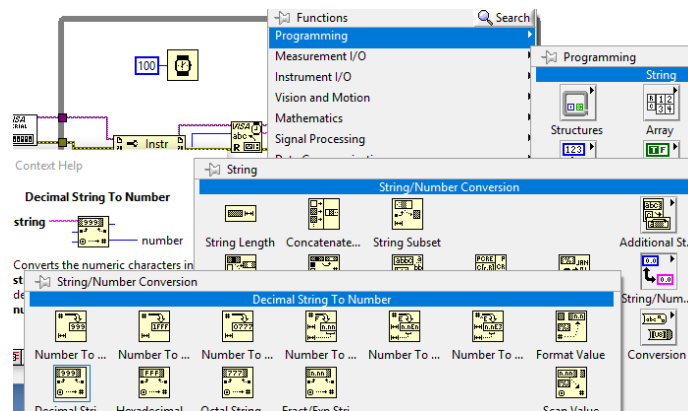
Συνδέουμε και το μηχανισμό Indicator για να φαίνονται οι χαρακτήρες που έρχονται από τη σειριακή επικοινωνία. Η σύνδεση τέτοιων μηχανισμών είναι η γνωστή διαδικασία του LabView όπως φαίνεται παρακάτω.



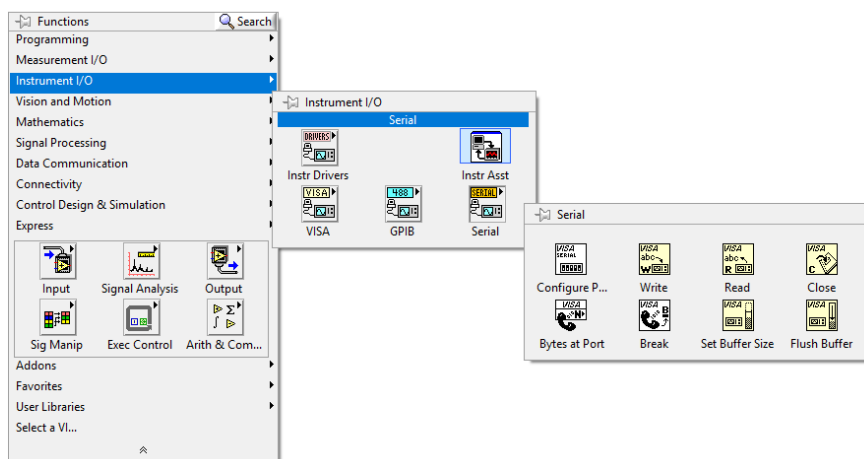
String to Byte Array: Μηχανισμός που ξεχωρίζει το μήνυμα σε διαδοχικούς ASCII χαρακτήρες και τους απεικονίζει σε πίνακα (array) με τη μορφή του δεκαδικού τους ισοδύναμου. Βρες τον πίνακα χαρακτήρων ASCII και πως αντιστοιχούν σε δυαδική, δεκαεξαδική και δεκαδική μορφή. Ο μηχανισμός βρίσκεται από τη διαδρομή που φαίνεται παρακάτω.



Decimal String to Number: μηχανισμός που μετατρέπει το μήνυμα που ήρθε με τη μορφή χαρακτήρων σε αριθμητική τιμή. Είναι απαραίτητη αυτή η μετατροπή ώστε να μπορούμε να διαχειριστούμε και να απεικονίσουμε τα δεδομένα της μέτρησης που έρχεται από το Arduino. Ο μηχανισμός βρίσκεται από τη διαδρομή που φαίνεται παρακάτω.

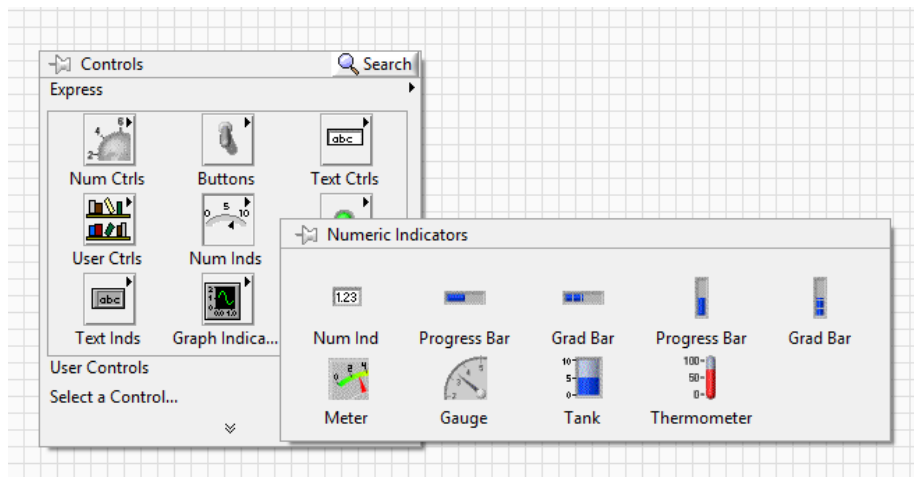


VISA Close: μηχανισμός που ολοκληρώνει τη σειριακή σύνδεση του εικονικού οργάνου (VI) του LabView με το Arduino μέσω σειριακής επικοινωνίας και με το καλώδιο USB.



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.

Thermometer, Progress Bar και άλλοι μηχανισμοί που χρησιμοποιούνται για την απεικόνιση δεδομένων βρίσκονται κάνοντας δεξί κλικ στο παράθυρο Front Panel/Controls όπως φαίνεται στην παρακάτω εικόνα.



ΚΑΛΗ ΕΠΙΤΥΧΙΑ