

Φύλλο εργασίας 3 – Αποθήκευση / Διάβασμα αρχείων

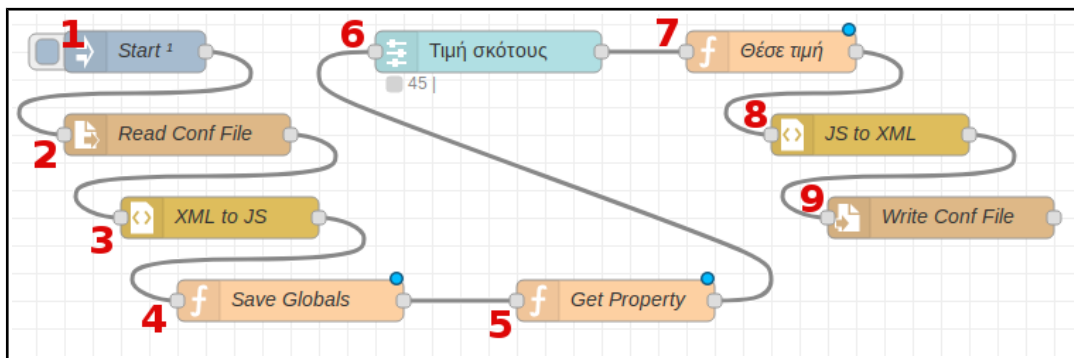
Στο προηγούμενο φύλλο βάλαμε ένα slider με το οποίο ρυθμίζουμε το κατώφλι της έντασης του ηλιακού φωτός. Αν η αναλογική τιμή πέσει κάτω απ' αυτό, τότε τα φώτα ανάβουν. Όλα λειτουργούν σωστά μέχρι να γίνει η πρώτη επανεκκίνηση του Raspberry Pi. Ενώ τα topics με τιμές Retain έχουν αποθηκευτεί από τον broker, η τιμή και επομένως η θέση του slider έχει χαθεί γιατί δεν έχει σχέση με κάποιο topic αλλά είναι μια καθολική μεταβλητή. Για να φορτώνουμε στο ξεκίνημα κάποιες αρχικές τιμές ρυθμίσεων θα πρέπει να χρησιμοποιήσουμε αρχεία.

Κατά το ξεκίνημα της υπηρεσίας Node-Red **διαβάζουμε** το αρχείο και αποθηκεύουμε τις τιμές του σε μεταβλητές με εμβέλεια σε όλο το **flow** ή σε **καθολικές** με εμβέλεια σε όλα τα flows.

Αν κατά την λειτουργία αλλάξει κάποια τιμή από αυτές που διαβάζονται από το αρχείο, τότε θα πρέπει να γίνει **εγγραφή** στο αρχείο ώστε να έχουμε μόνιμη αποθήκευση της τιμής.

Διάγραμμα ροής

Σχεδιάζουμε το παρακάτω διάγραμμα και εξηγούμε ένα-ένα τα blocks.



Ανάγνωση από αρχείο

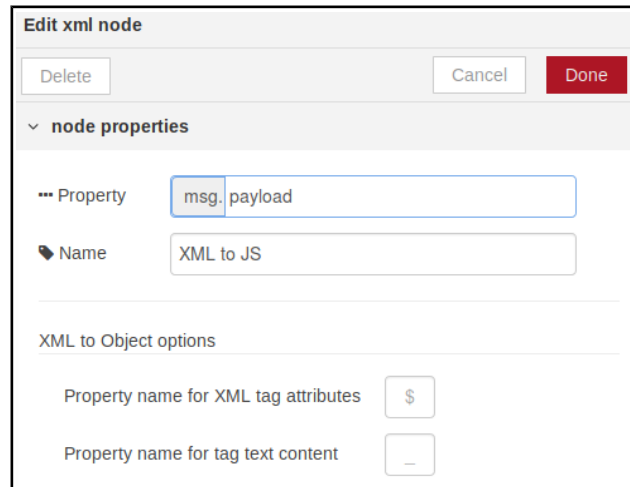
1. Από την ομάδα **input** βάζουμε ένα block **inject**. Στο payload βάζουμε τύπο Number χωρίς περιεχόμενο και κάνουμε tick το Inject once after 0.1 seconds, then στο Repeat το αφήνουμε None. Με αυτό τον τρόπο εκτελούμε τα επόμενα blocks μια φορά κατά την εκκίνηση του Node-Red.
2. Από την ομάδα **storage** βάζουμε το block **file in**. Το αρχείο είναι αποθηκευμένο στο home (~) του user pi.

The screenshot shows the 'Edit file in node' dialog box. It has buttons for 'Delete', 'Cancel', and 'Done'. Under 'node properties', there are fields for 'Filename' (containing 'conf.txt'), 'Output' (a dropdown menu showing 'a single utf8 string'), a checkbox for 'Send message on error (legacy mode)', and a 'Name' field (containing 'Read Conf File'). A tip at the bottom states: 'Tip: The filename should be an absolute path, otherwise it will be relative to the working directory of the Node-RED process.'

3. Ο τύπος του αρχείου θα είναι XML επομένως θέλουμε να γίνει μετατροπή από XML σε Java Script. Από την ομάδα **functions** εισάγουμε το block **xml**. Αυτό το block λειτουργεί αμφίδρομα και στην συγκεκριμένη περίπτωση, εφόσον διαβάσει XML μας επιστρέφει ένα αντικείμενο Java Script `{"settings":{"gard_slider_val":["45"],"const1":["3"],"const34":["7"]}}`.

Το αρχείο XML του παραδείγματος είναι κάπως έτσι:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<settings>
  <gard_slider_val>45</gard_slider_val>
  <const1>3</const1>
  <const34>7</const34>
</settings>
```



4. Από την ομάδα **function** βάζουμε το block **function**. Εδώ φτιάχνουμε μια δική μας συνάρτηση σε Java Script η οποία διαβάζει το αντικείμενο λεξικού και εξάγει τα ζεύγη για το κλειδί settings. Τα ζεύγη τα **αποθηκεύουμε** για καθολική χρήση απ' όλο το flow. Στην έξοδο αυτού του μπλοκ θα μπορούσαμε να συνδέσουμε πολλές συναρτήσεις όπως η επόμενη (5) ώστε να εξάγουμε πολλά ζεύγη από το λεξικό.

```
1 //Ο προηγούμενος στέλνει αντικείμενο
2 settings = msg.payload['settings']; //Πάρε από τον προηγούμενο το λεξικό settings
3 flow.set('settings', settings); //Κάντο καθολικό για χρήση από άλλους κόμβους
4 msg.payload = settings; //Αλλάξε το payload με το λεξικό
5 return msg; //Επέστρεψε το λεξικό στον επόμενο
```

5. Πάλι βάζουμε μια **custom** συνάρτηση προκειμένου να εξάγουμε την τιμή του κλειδιού που μας ενδιαφέρει. Μπορούμε να έχουμε πολλές ίδιες αλλά με διαφορετικά κλειδιά.

```
1 settings = msg.payload; //Είναι το λεξικό από τον προηγούμενο
2 val = settings['gard_slider_val']; //Πάρε την τιμή που σε ενδιαφέρει
3 msg.payload = parseInt(val); //Να γίνει ακέραια
4 return msg; //Επέστρεψε στο επόμενο την ακέραια τιμή
```

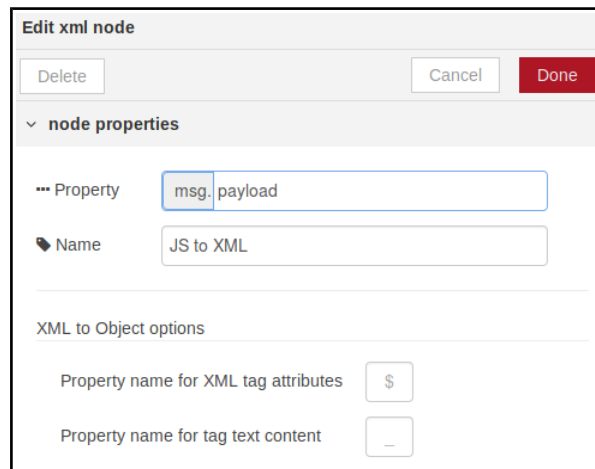
6. Το slider είναι ακριβώς το ίδιο μ' αυτό στο προηγούμενο φύλλο. Η ακέραια τιμή πηγαίνει στην είσοδο ώστε να τοποθετείται ο δρομέας στην θέση του κατά το ξεκίνημα σύμφωνα με την τιμή που βρήκε στο αρχείο.

Εγγραφή σε αρχείο

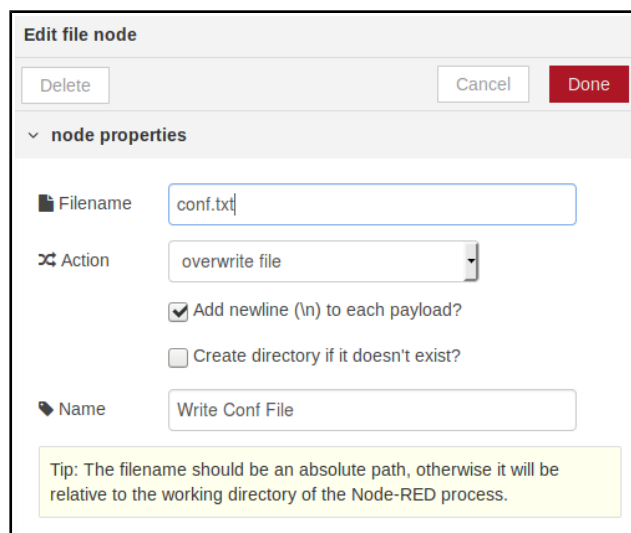
7. Αν ο χρήστης σύρει το slider, η αλλαγή της τιμής θα ενεργοποιήσει την συνάρτηση 'Θέσε τιμή' που ακολουθεί ο κώδικας.

```
1 var settings = flow.get('settings') || {}; //Πάρε καθολικές τιμές του λεξικού
2 gard_intens = msg.payload; //Η τιμή από τον προηγούμενο κόμβο
3 //val = {gard_slider_val: gard_intens.toString()};
4 settings["gard_slider_val"] = gard_intens.toString(); //Βάλε στο κλειδί την τιμή
5 msg.payload = {settings: settings}; //Αλλάξε το payload, το root να γίνει 'settings'
6 flow.set('settings', settings); //Αποθήκευσε τις αλλαγές για χρήση από άλλους κόμβους
7 return msg; //Επέστρεψε μήνυμα στον επόμενο
```

8. Ακολουθεί το ίδιο μπλοκ με το 3 και κάνει μετατροπή από Java Script σε XML. Στην είσοδο αυτού του μπλοκ μπορεί να συνδεθούν πολλά παρόμοια με το παραπάνω (7) από άλλα controls του flow.



9. Τέλος από την ομάδα **storage** επιλέγουμε το μπλοκ **file** το οποίο γράφει σε αρχείο.



Τέλος πρέπει να γίνει μια μικρή τροποποίηση στη συνάρτηση 'Ενέργεια' του προηγούμενου φύλλου ώστε να διαβάζει από το λεξικό settings την τιμή του κλειδιού gard_slider_val.

```
var payload = msg.payload;
var state = context.get('state') || false;
var settings = flow.get('settings') || {};
var gard_intens = parseInt(settings["gard_slider_val"]);
if (payload < gard_intens)
  if (!state)
  {
    msg.payload = "ON";
    state = true;
  }
  else
    msg.payload = "";
else
  if (state)
  {
    msg.payload = "OFF";
    state = false;
  }
  else
    msg.payload = "";
context.set('state', state);
if (msg.payload !== "")
  return msg;
```