

# **Σύστημα Ευφυούς Άρδευσης (Smart Pot) Arduino + LabView**

**Αχιλλέας Παππάς  
Δημήτρης Δούπας  
Χρήστος Ζαρουτιέρης**

Ιανουάριος 2026

—

Τμήμα Μηχανικών Παραγωγής και  
Διοίκησης – Διεθνές Πανεπιστήμιο  
Ελλάδος

—

Μ102 – Συλλογή και Επεξεργασία  
Δεδομένων

---

# Εισαγωγή

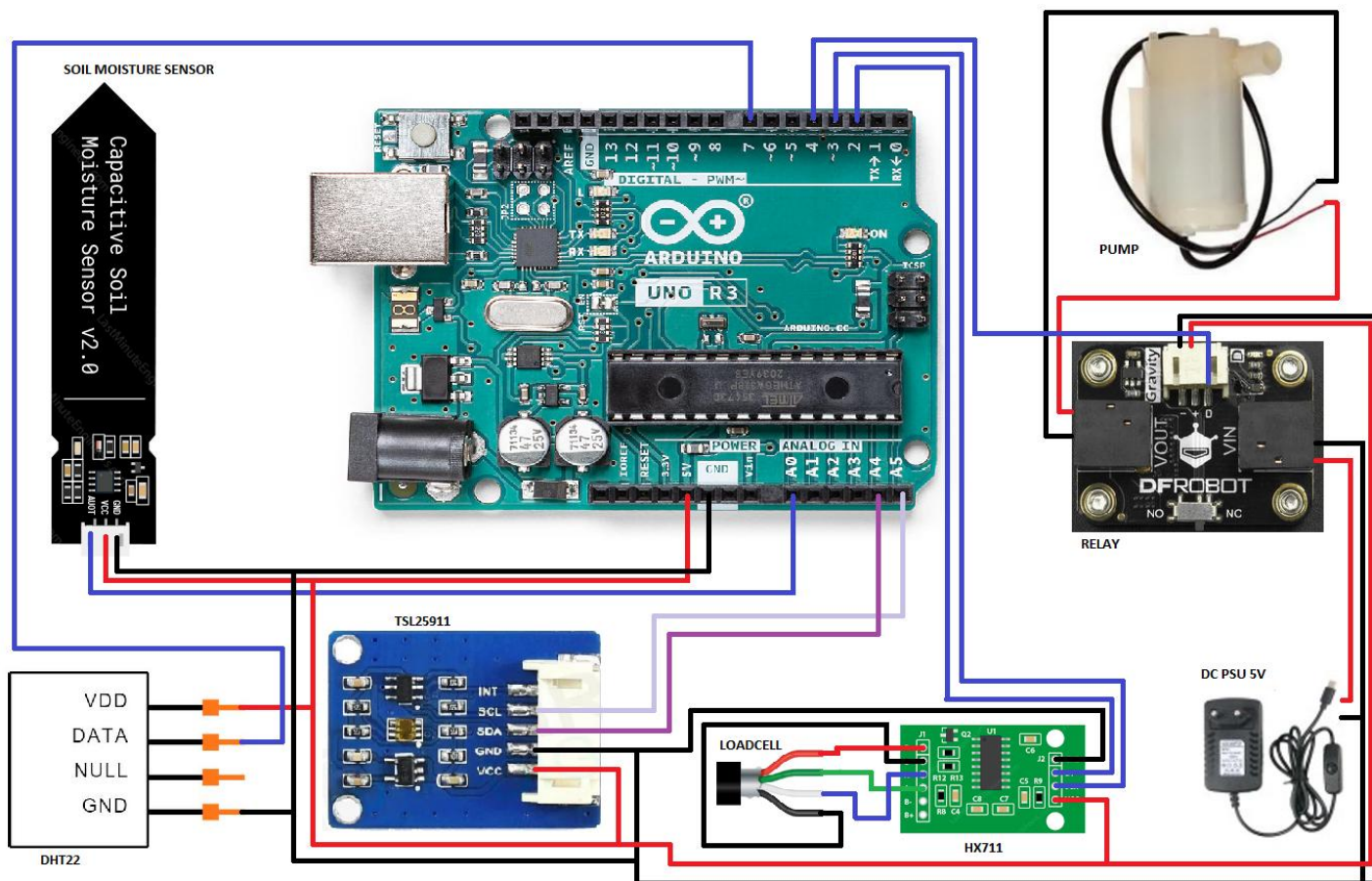
Σκοπός της εργασίας είναι η ανάπτυξη ενός απλού συστήματος «ευφυούς γεωργίας» σε κλίμακα γλάστρας (smart pot), με δυνατότητες συνεχούς καταγραφής περιβαλλοντικών μετρήσεων και αξιοποίησής τους για αυτόματη άρδευση ενός φυτού.

Η υλοποίηση οργανώνεται γύρω από το LabVIEW, το οποίο λειτουργεί ως περιβάλλον εποπτείας (supervisory control): προβάλλει σε πραγματικό χρόνο τις μετρήσεις στο Front Panel και υλοποιεί τη λογική απόφασης για πότισμα. Το Arduino Uno χρησιμοποιείται ως διεπαφή με τον φυσικό κόσμο, αναλαμβάνοντας την ανάγνωση των αισθητήρων και τον έλεγχο του ενεργοποιητή άρδευσης (αντλία μέσω relay).

Η επικοινωνία Arduino–LabVIEW βασίζεται σε USB serial με καθορισμένο baud rate και συμφωνημένο σχήμα αποστολής δεδομένων (γραμμή CSV). Η επιλογή αυτή προτιμάται, γιατί επιτρέπει ευέλικτη μορφοποίηση των μετρήσεων στο Arduino και απλή αποσυσκευασία στο LabVIEW, χωρίς εξάρτηση από έτοιμα toolkits που δεν καλύπτουν πάντα βιβλιοθήκες όπως DHT22 ή HX711.

## Hardware

Το υλικό της υλοποίησης χτίζεται γύρω από Arduino Uno και επιλέγεται με κριτήρια χαμηλού κόστους, διαθεσιμότητας και καθαρής διασύνδεσης.



## Αισθητήρες:

- i. Waveshare Ambient Light Sensor (TSL2591/TSL25911 μέσω I2C, SDA/SCL)
- ii. DHT22 για θερμοκρασία και σχετική υγρασία (ψηφιακή γραμμή δεδομένων)
- iii. Gravity capacitive soil moisture sensor με αναλογική έξοδο (π.χ. Α0).

## Άρδευση:

- Υποβρύχια αντλία 5V που τροφοδοτείται από εξωτερικό τροφοδοτικό και ενεργοποιείται μέσω Relay Module.
- Ποσοτικός έλεγχος ποτίσματος: load cell 5 kg σε συνδυασμό με HX711 (DT/DOUT και SCK/CLK) ώστε το πότισμα να τερματίζει όταν επιτευχθεί αύξηση βάρους ίση με τη στοχευμένη ποσότητα νερού.

## Πρακτικές παραδοχές:

Κοινή γείωση μεταξύ Arduino και τροφοδοσίας της αντλίας για αξιόπιστο έλεγχο του relay, απομάκρυνση καλωδίων ισχύος από γραμμές HX711/load cell και (όπου χρειάζεται) απλό φιλτράρισμα στη ζύγιση (π.χ. μέσος όρος πολλαπλών δειγμάτων).

# Arduino IDE

## Arduino Module:

**Το Arduino αναλαμβάνει:** (1) **αρχικοποίηση** και ανάγνωση DHT22, TSL2591/TSL25911, soil moisture (ADC) και (αν χρησιμοποιείται) HX711, (2) **μορφοποίηση** και **αποστολή μετρήσεων** σε γραμμή CSV, (3) λήψη εντολής από το LabVIEW για **ενεργοποίηση/απενεργοποίηση ποτίσματος**, και (4) ασφαλή εκτέλεση του ποτίσματος μέσω relay.

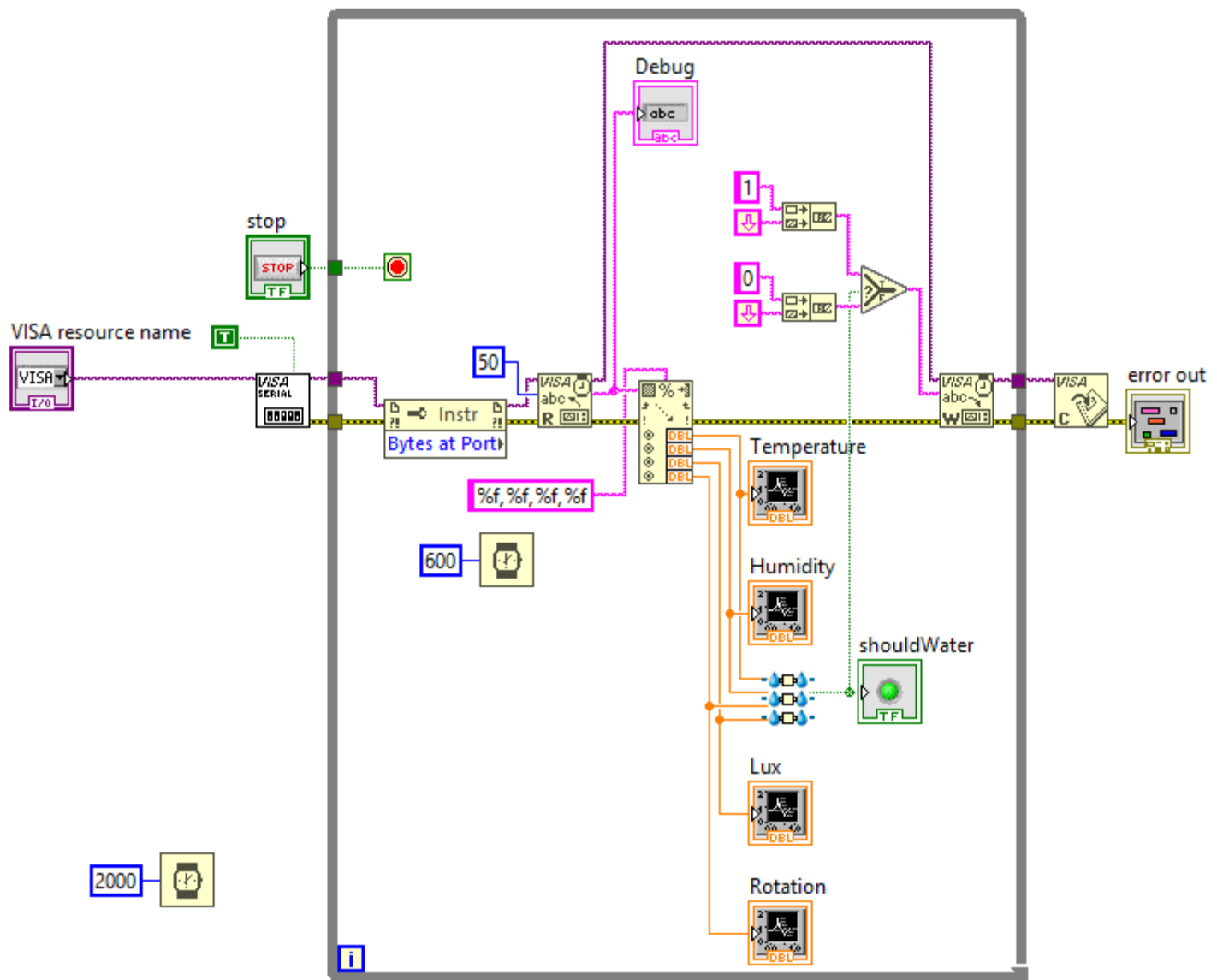
## Λογική Ποτίσματος:

Η λογική ποτίσματος υλοποιείται ως **state machine** με βασικές καταστάσεις IDLE/WATERING. Όταν δοθεί εντολή για πότισμα, καταγράφεται αρχικό βάρος (tare/average), ορίζεται βάρος-στόχος (initial + Δwater) και ενεργοποιείται η αντλία. Το πότισμα τερματίζει είτε όταν η ένδειξη βάρους φτάσει το στόχο είτε όταν συμπληρωθεί timeout ασφαλείας.

Σχετικά Αρχεία: [hwCode.ino](#)

# Κεφάλαιο 4 – LabVIEW

Το LabVIEW αποτελεί το κεντρικό σημείο ελέγχου. Το βασικό VI εκτελείται σε While Loop μέχρι να πατηθεί το Stop, και υλοποιεί σταθερή αλληλουχία: READ → PARSE → UPDATE UI → LOG → DECIDE → SEND CMD.



## Σειριακή επικοινωνία (VISA):

Χρησιμοποιείται VISA Configure Serial Port με επιλογή VISA resource name (COM). Στη συνέχεια, το Bytes at Port χρησιμοποιείται ως έλεγχος διαθεσιμότητας δεδομένων. Με βάση το block diagram, το VISA Read διαβάζει ένα σταθερό chunk (π.χ. 50 bytes). Η επιλογή είναι λειτουργική εφόσον η γραμμή CSV είναι μικρότερη από το chunk και αποστέλλεται συχνά.

## Parsing:

Η εισερχόμενη συμβολοσειρά αποσυνσκευάζεται με Scan From String και format "%f,%f,%f,%f" σε τέσσερις DBL μεταβλητές. Οι τιμές δρομολογούνται σε αριθμητικές ενδείξεις και γραφήματα (Temperature, Humidity, Lux/light level και Soil Moisture). Παράλληλα, κρατιέται το raw string σε Debug indicator ώστε να φαίνεται άμεσα αν υπάρχουν κομμένες/ελλιπείς γραμμές.

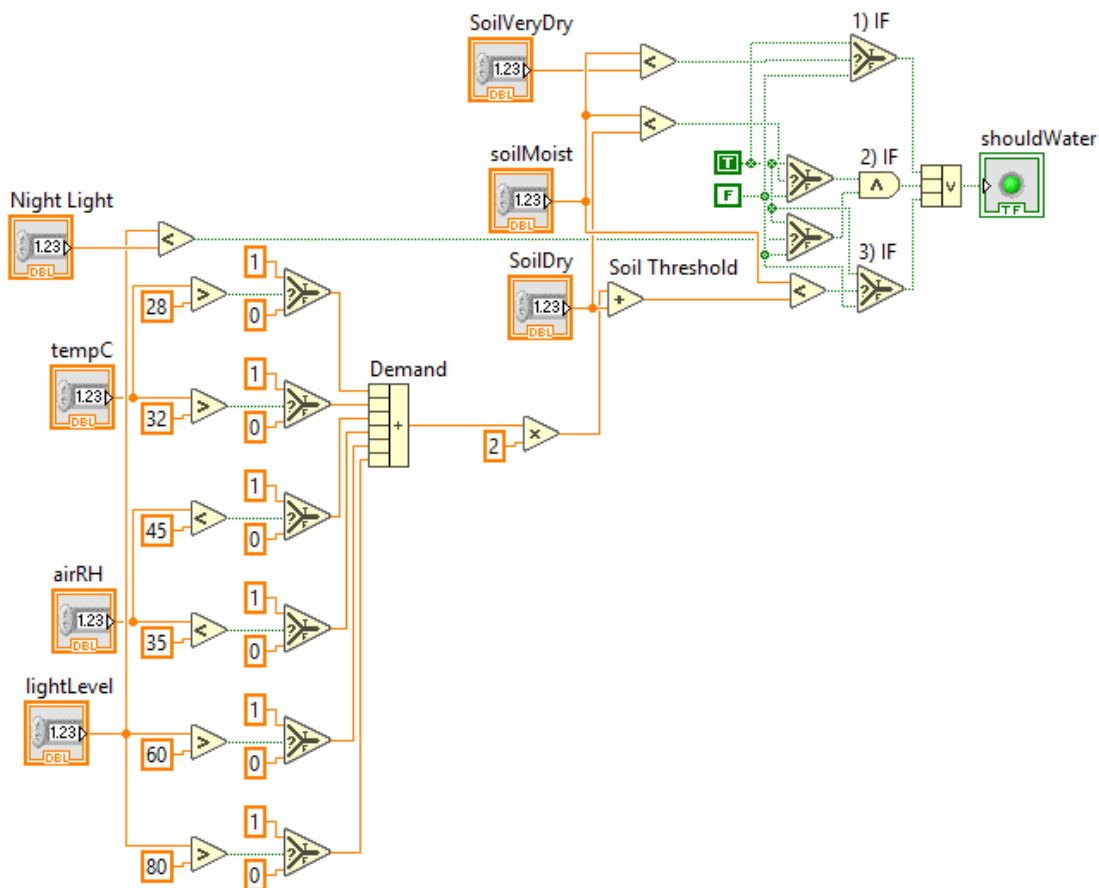


## Ρυθμός λειτουργίας:

Ένα Wait (ms) (π.χ. 600 ms) σταθεροποιεί τον ρυθμό δειγματοληψίας. Το επιλέξαμε αυθαίρετα με την προϋπόθεση ότι δεν ήταν διαιρέτης με το 1000ms που είναι ο ρυθμός που στέλνει δεδομένα το Arduino για την αποφυγή προβλημάτων συγχρονισμού. Παράλληλα βάζουμε χρόνο αναμονής 2000ms πριν το loop ώστε να αρχικοποιηθεί η επικοινωνία μέσω serial.

## Λογική απόφασης shouldWater:

Η συνάρτηση shouldWater(tempC, airRH, soilMoist, lightLevel) παράγει boolean έξοδο. Στο δικό μας μοντέλο, το soilMoist λειτουργεί ως βασικό κριτήριο, ενώ θερμοκρασία, υγρασία αέρα και φως συνεισφέρουν σε demand score που μετατοπίζει δυναμικά το threshold. Επιπλέον εφαρμόζεται συντηρητικός κανόνας για «night mode» (χαμηλό lightLevel). Αφού σχεδιάσαμε τον ψευδοκώδικα, υλοποιήσαμε την λογική σε δικό της VI. Αξίζει να σημειωθεί ότι οι αρχικές τιμές των SoilMoist/dry δίνονται στο front panel και τίθενται ως default για την απλούστευση του τελικού κώδικα.



## Επιστροφή εντολής στο Arduino:

Το boolean `shouldWater` μετατρέπεται σε κείμενο βάσει boolean (1/0) με Select, και αποστέλλεται με VISA Write μέσω της ίδιας σειριακής σύνδεσης. Το LabVIEW δηλαδή αποφασίζει (supervisory control), ενώ το Arduino εκτελεί και εφαρμόζει ασφαλιστικές δικλείδες (στόχος βάρους/timeout).

## Διαχείριση σφαλμάτων:

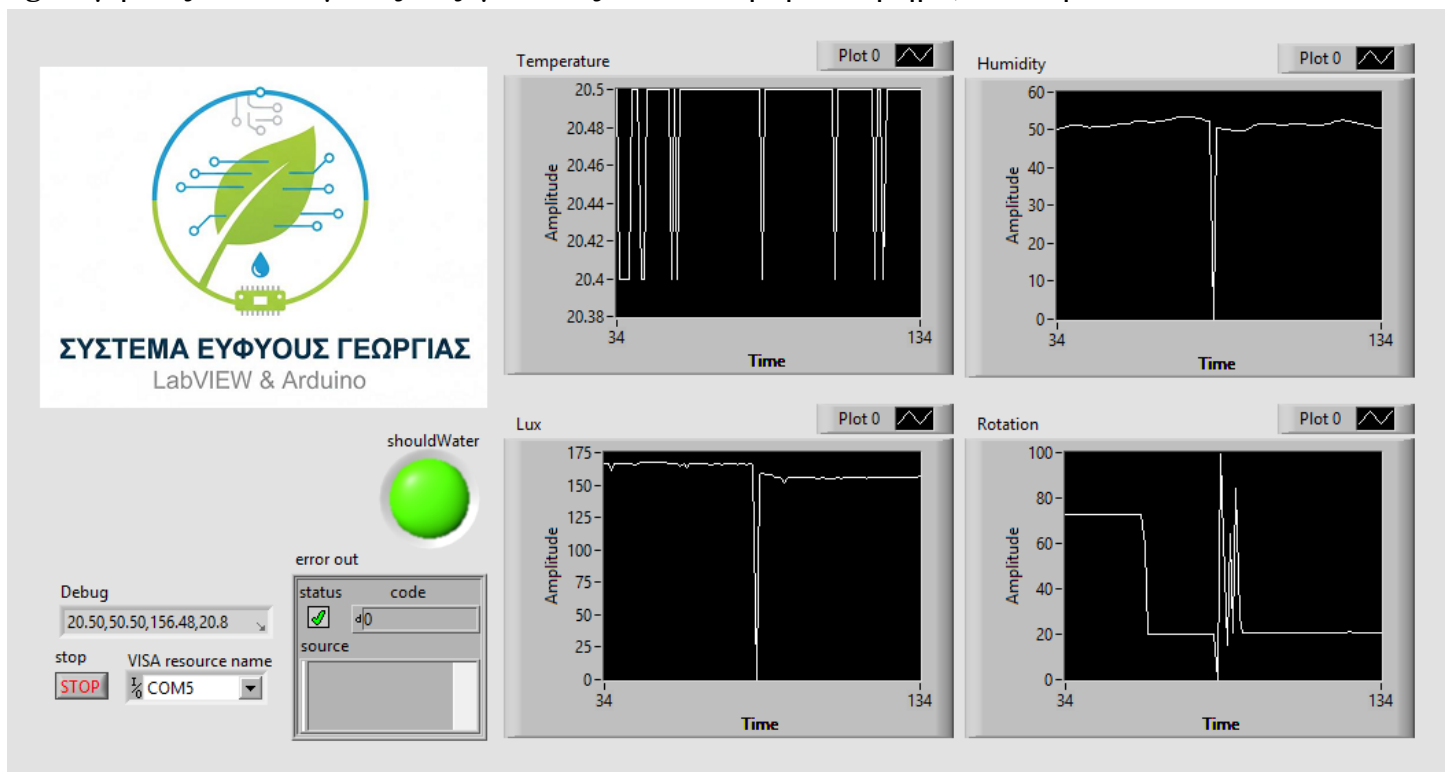
Η αλυσίδα error cluster από VISA nodes επιτρέπει ανίχνευση αποτυχίας σύνδεσης/ανάγνωσης. Στην πράξη, ένα απλό Case μπορεί να σταματά το loop ή να μηδενίζει την εντολή ποτίσματος όταν υπάρχει VISA error ή όταν δεν λαμβάνονται δεδομένα για συγκεκριμένο χρονικό διάστημα (watchdog).

Σχετικά Αρχεία: [irrigationSystem.vi](#), [shouldWater.vi](#), [shouldWaterAlgorithm.txt](#)

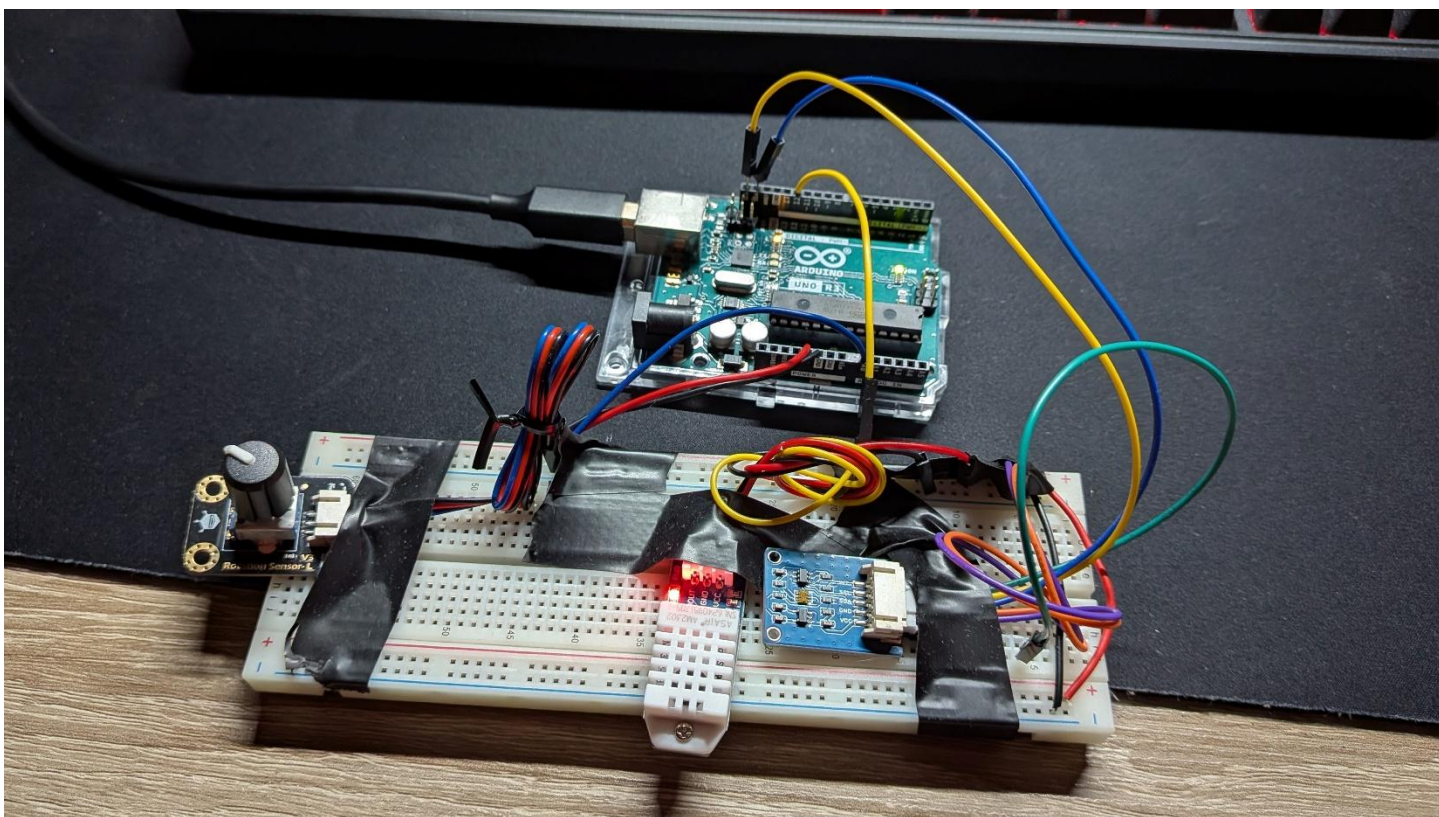
## Πειραματική αξιολόγηση

Η αξιολόγηση έχει χαρακτήρα **proof-of-concept** και στοχεύει να τεκμηριώσει ότι το σύστημα λειτουργεί ως ενιαίο σύνολο: Arduino (μέτρηση/εκτέλεση) → serial → LabVIEW (απεικόνιση/απόφαση) → serial → Arduino (ενέργεια).

Στο Front Panel χρησιμοποιούνται βασικά γραφήματα για **Temperature**, **Humidity**, **soil moisture** και Lux. Αναμενόμενα, οι μεταβλητές αέρα μεταβάλλονται πιο ομαλά, ενώ soil moisture και light εμφανίζουν πιο άμεσες αυξομειώσεις (τοποθέτηση αισθητήρα, σκίαση).



Για την παρακολούθηση της ανταπόκρισης του labview, χρησιμοποιήσαμε ένα ποτενσιόμετρο για την είσοδο του soil moisture. Όταν το soil moisture έπεφτε κάτω από το όριο, το should water γινόταν 1 και το Arduino λάμβανε το input και εκτελούσε τις πειραματικές ενέργειες (άνοιγμα onboard led). Στο πραγματικό σύστημα θα ενεργοποιούσε το ρελέ του ποτίσματος και θα παρακολουθούσε το βάρος της γλάστρας.



Σχετικά Αρχεία: [hwTestCode.ini](#)

## Συζήτηση

**Το πρωτότυπο αναδεικνύει την πρακτική αξία του LabVIEW ως περιβάλλον εποπτείας:** μέσα στο ίδιο VI συνυπάρχουν real-time απεικόνιση και λογική ελέγχου, με διαφανή ροή δεδομένων.

**Περιορισμοί:** εξάρτηση από PC που εκτελεί LabVIEW και από σταθερή serial σύνδεση, ανάγκη βαθμονόμησης αισθητήρων (ειδικά soil moisture), ευαισθησία ζύγισης σε κραδασμούς/θόρυβο όταν λειτουργεί η αντλία, και χρήση απλού rule-based μοντέλου για την απόφαση.

**Ρεαλιστικές επεκτάσεις:** για αυτόνομη χρήση πεδίου, η λογική απόφασης μπορεί να μεταφερθεί σε embedded πλατφόρμα (π.χ. ESP32/Raspberry Pi headless) με ασύρματη επικοινωνία και το LabVIEW να παραμείνει ως εργαλείο ανάπτυξης/δοκιμών ή ως κονσόλα εποπτείας.

**LabVIEW σε smart agriculture:** χρησιμοποιείται συχνά σε πιλοτικές εγκαταστάσεις/ερευνητικά πρωτόκολλα όπου προέχει η γρήγορη ανάπτυξη, η μετρητική τεκμηρίωση και η ευκολία αλλαγής κανόνων. Σε κλιμάκωση πολλών κόμβων χαμηλού κόστους, η αρχιτεκτονική τείνει να μεταφέρεται σε distributed embedded nodes, με το LabVIEW (αν υπάρχει) να παίζει ρόλο συγκέντρωσης/ανάλυσης.