**Πανεπιστήμιο Θεσσαλίας**

**Σχολή Θετικών Επιστημών**

**Τμήμα Πληροφορικής με Εφαρμογές στη Βιοϊατρική**

**Αναφορά Πτυχιακής Εργασίας**

**ΣΧΕΔΙΑΣΗ ΣΥΣΤΗΜΑΤΟΣ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΦΥΣΙΚΩΝ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΓΙΑ ΤΗΝ ΑΣΦΑΛΕΙΑ ΙοΤ ΣΥΣΚΕΥΩΝ**

**Ελένη Σουλίδου**

**Λαμία, 2024**

**ΠΕΡΙΕΧΟΜΕΝΑ**

[**ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΥΚΛΩΜΑΤΟΣ** 4](#_Toc164722072)

[**Raspberry Pi Zero W** 4](#_Toc164722073)

[**Arduino UNO R3** 4](#_Toc164722074)

[**MCP9808** 5](#_Toc164722075)

[**ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ** 6](#_Toc164722076)

[**Παρουσίαση Συστήματος** 6](#_Toc164722077)

[**Κώδικας Υλοποίησης** 6](#_Toc164722078)

[**Μετρήσεις και Πειραματική Αξιολόγηση** 6](#_Toc164722079)

**ΕΙΣΑΓΩΓΗ**

Στην παρούσα αναφορά θα αναλύσουμε συνοπτικά τα χαρακτηριστικά των συνιστωσών της υλοποίησης μας, τον τρόπο διασύνδεσης και λειτουργίας τους, καθώς και τον κώδικα που απαιτείται για τον προγραμματισμό τους. Αρχικά θα ξεκινήσουμε με μια σύντομη ανάλυση των επιμέρους στοιχείων της υλοποίησης. Στη συνέχεια, θα παρουσιαστεί η κατασκευή και η διασύνδεση των επιμέρους στοιχείων, ο κώδικας και οι μετρήσεις. Τέλος, θα γίνει και μια σύντομη βιβλιογραφική ανασκόπηση, ερευνητικών εργασιών με συναφή θεματολογία.

# **ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΥΚΛΩΜΑΤΟΣ**

Το παρόν κύκλωμα αποτελείται από έναν μικροϋπολογιστή Raspberry Pi Zero W, έναν μικροελεγκτή Arduino UNO και έναν αισθητήρα θερμοκρασίας MCP9808, τα οποία και θα αναλύσουμε.

### **Raspberry Pi Zero W**

Ο μικροϋπολογιστής συνοδεύεται από έναν μονοπύρηνο επεξεργαστή BCM2835, αρχιτεκτονικής 32 bit, χρονισμένο στο 1GHz και από μνήμη RAM χωρητικότητας 512MB. Χρησιμοποιώντας την θύρα CSI που μας παρέχει ο μικρουπολογιστής, έχουμε συνδέσει μια κάμερα ανάλυσης 5MP ώστε να λειτουργεί σαν IP κάμερα για τις ανάγκες της υλοποίησης μας. Για την διασύνδεση της υλοποίησης με το υπόλοιπο δίκτυο, ο μικροϋπολογιστής διαθέτει διεπαφή πρωτοκόλλου IEEE 802.11b/g/n (WiFi) με συχνότητα λειτουργίας στα 2.4 GHz.

Η εκτέλεση των προγραμμάτων πάνω στο Raspberry Pi Zero W υποστηρίζεται από το λειτουργικό σύστημα Raspberry Pi OS Legacy Lite 32-bit. Η έκδοση Lite του λειτουργικού συστήματος αποτελεί μια μινιμαλιστική εικόνα λογισμικού, αποτελούμενη από 493 πακέτα, από την οποία λείπει ο διαχειριστής X-window. Λόγω αυτού, το σύστημα είναι ταχύτερο και πιο συμβατό με περιβάλλοντα εξυπηρετητών και Internet of Things. (paper me RPi).

### **Arduino UNO R3**

Το σύστημα του Arduino βασίζεται στον μικροελεγκτή ATmega328P, ο οποίος λειτουργεί ως κύριος επεξεργαστής χρονισμένος στα 16MHz, ενώ χρησιμοποιεί και τον επεξεργαστή ATmega16U2 για τη μετατροπή των σημάτων USB από τον υπολογιστή σε σειριακά σήματα που μπορεί να κατανοήσει ο ATmega328P. Ουσιαστικά, λειτουργεί ως γέφυρα μεταξύ της θύρας USB του υπολογιστή και του μικροελεγκτή. Επίσης, περιέχει μνήμες SRAM χωρητικότητας 2KB, FLASH χωρητικότητας 32KB και EEPROM χωρητικότητας 1KB.

Ως κομμάτι της υλοποίησης, το σύστημα του Arduino πραγματώνει το σύστημα παρακολούθησης (monitor system) σε σύνδεση με τον αισθητήρα θερμοκρασίας MCP9808. Κύρια εργασία του Arduino είναι η συλλογή των δεδομένων από τον αισθητήρα. Οι μετρήσεις που λαμβάνονται από το Arduino, καταγράφονται από το πρόγραμμα RealTerm σε αρχεία κειμένου.

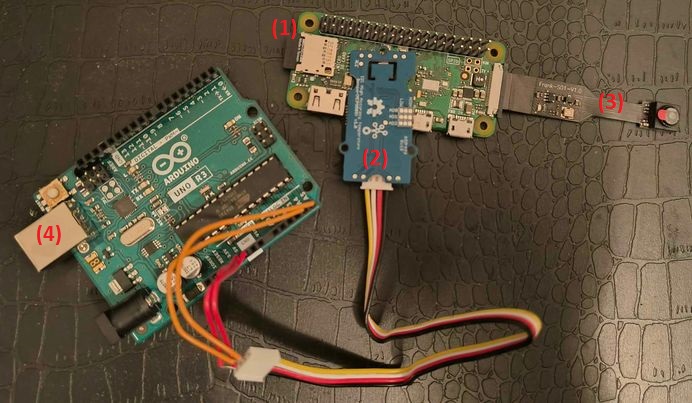
### **MCP9808**

Είναι ένας ψηφιακός αισθητήρας θερμοκρασίας ο οποίος μετατρέπει θερμοκρασίες μεταξύ - 20℃ έως +100℃ σε μια ψηφιακή λέξη με ακρίβεια ±0,5℃. Επίσης διαθέτει τέσσερις διαφορετικές αναλύσεις μέτρησης (+0,5℃, +0,25℃, +0,125℃ και +0,0625℃). Ο MCP9808 είναι συμβατός με το πρωτόκολλο επικοινωνίας I2C, το οποίο χρησιμοποιεί μόνο δύο ακροδέκτες του μικροελεγκτή για την μεταφορά δεδομένων, πέραν των ακροδεκτών για το ρεύμα και την γείωση, τον Serial Data Line (SDA) ο οποίος είναι ο Α4 και τον Serial Clock Line (SCL) ο οποίος είναι ο Α5.

# **ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ**

### **Παρουσίαση Συστήματος**

Το σύστημα της υλοποίησης, καθώς και η διασύνδεση των επιμέρους στοιχείων φαίνονται στην παρακάτω εικόνα.



Εικόνα 1: Απεικόνιση συστήματος παρακολούθησης θερμοκρασίας.

Το σύστημα κατασκευάζεται με βάση την λήψη μετρήσεων θερμοκρασίας από την ΚΜΕ του Raspberry Pi (1). Ο αισθητήρας θερμοκρασίας MCP9808 (2) έχει τοποθετηθεί πάνω από τον επεξεργαστή του Raspberry Pi με τη βοήθεια θερμοαγώγιμης πάστας, για την ανίχνευση των μεταβολών της θερμοκρασίας κατά την λειτουργία του. Με την σειρά του, ο αισθητήρας συνδέεται στους ακροδέκτες A4 και A5 του Arduino UNO R3 (4) για την μεταφορά των μετρήσεων στον μικροελεγκτή. Το Raspberry Pi συνδέεται με την κάμερα (3), λειτουργώντας ως IP Camera.

### **Κώδικας Υλοποίησης**

Η παραπάνω διασύνδεση (Εικόνα 1) υποστηρίζεται από τον μικροελεγκτή, πάνω στον οποίο γράφεται ο κώδικας ελέγχου του συστήματος. Συγκεκριμένα, ο κώδικας αφορά την λειτουργία του αισθητήρα θερμοκρασίας.

#include <Wire.h>

#include "Adafruit\_MCP9808.h"

Adafruit\_MCP9808 tempsensor = Adafruit\_MCP9808();

void setup() {

  Serial.begin(9600);

  while (!Serial);

  Serial.println("MCP9808 temp sensor");

  if (!tempsensor.begin(0x18)) {

    Serial.println("Couldn't find MCP9808! Check your connections and verify the address is correct.");

    while (1);

  }

   Serial.println("Found MCP9808!");

  tempsensor.setResolution(0);

}

void loop() {

  float c = tempsensor.readTempC();

  Serial.print(c, 2);

  Serial.println("\t");

  delay(100);

}

Η βιβλιοθήκη Wire.h αξιοποιείται για την επικοινωνία του μικροελεγκτή με συσκευές που χρησιμοποιούν το I2C πρωτόκολλο. Με την σειρά της, η βιβλιοθήκη Adafruit\_MCP9808.h περιέχει όλες τις συναρτήσεις που είναι απαραίτητες για να διαβάσουμε δεδομένα από τον αισθητήρα. Για την αναπαράσταση του αισθητήρα στον κώδικα δημιουργούμε ένα αντικείμενο κλάσης Adafruit\_MCP9808(), το οποίο και θα χρησιμοποιούμε από εδώ και στο εξής. Μέσα στη συνάρτηση setup() δηλώνουμε ότι θα χρησιμοποιήσουμε το Serial terminal με ρυθμό baud 9600 και περιμένουμε να ξεκινήσει η μετάδοση, ενημερώνοντας με κατάλληλο μήνυμα.

Επειδή ο αισθητήρας χρησιμοποιεί το I2C πρωτόκολλο, χρειάζεται να του υποδείξουμε σε ποια διεύθυνση θα συνδεθεί, διότι παρέχεται η δυνατότητα να συνδεθούν πολλοί αισθητήρες στον ίδιο δίαυλο I2C χρησιμοποιώντας διαφορετική διεύθυνση ο καθένας. Έτσι, οι διευθύνσεις που μπορούν να χρησιμοποιηθούν είναι από το 0x18 μέχρι το 0x1F, με την διεύθυνση 0x18 να είναι η προεπιλεγμένη. Έπειτα, θέτουμε την ανάλυση μέτρησης του αισθητήρα, η οποία μπορεί να είναι μια από τις εξής επιλογές:

|  |  |  |
| --- | --- | --- |
| **Λειτουργία** | **Ανάλυση** | **Περίοδος δειγματοληψίας** |
| 0 | 0.5℃ | 30 ms |
| 1 | 0.25℃ | 65 ms |
| 2 | 0.125℃ | 130 ms |
| 3 | 0.0625℃ | 250 ms |

Τέλος, στη συνάρτηση loop() διαβάζουμε τις μετρήσεις από τον αισθητήρα σε βαθμούς Κελσίου και τις εμφανίζουμε στο Serial terminal ανά 100 ms.

### **Μετρήσεις και Πειραματική Αξιολόγηση**

Οι μετρήσεις πάρθηκαν σε διάφορες καταστάσεις, όπως κατά την διάρκεια εκκίνησης του συστήματος, κατά την κατάσταση αδράνειας του συστήματος και κατά την κανονική λειτουργία του συστήματος. Οι συγκεκριμένες μετρήσεις καταγράφηκαν σε ένα αρχείο και στη συνέχεια διασπάστηκαν στα επιμέρους διαγράμματα. Επίσης, το σύστημα δοκιμάστηκε σε διάφορες συνθήκες υπό το φως της ημέρες και αντίστοιχα στο σκοτάδι. Για την συνέχη ροή εικόνας από την κάμερα χρησιμοποιείται η εντολή raspivid που παρέχεται από το λειτουργικό σύστημα του Raspberry Pi και για την παρακολούθηση της ροής χρησιμοποιείται η εφαρμογή VLC. Η εντολή συντάσεται ως εξής: « raspivid -o - -t 0 -hf -w 800 -h 400 -fps 24 |cvlc -vvv stream:///dev/stdin --sout '#standard{access=http,mux=ts,dst=:8160}' :demux=h264 ».

**Παράμετροι:**

-o : Καθορίζει το όνομα αρχείου εξόδου. Το '-' δίπλα δηλώνει ότι δεν υπάρχει όνομα αρχείου.

-t : Είναι η διάρκεια της καταγραφής, με το 0 να σημαίνει άπειρο.

-hf : Ανατροπή ως προς τον οριζόντιο άξονα.

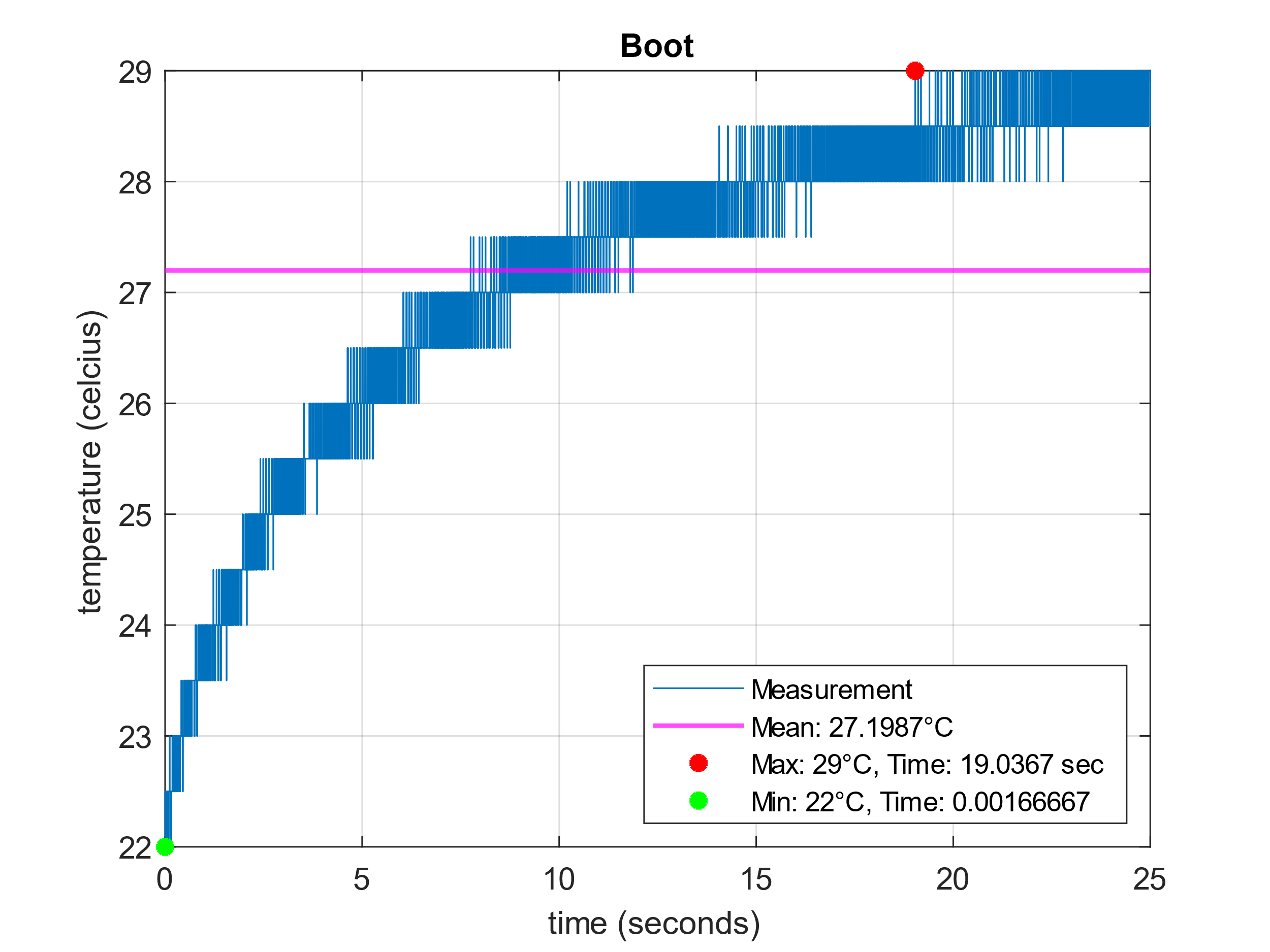
-w, -h : ορίζουν την ανάλυση για το πλάτος και το ύψος.

-fps : Ορίζει την συχνότητα καταγραφής της εικόνας.

Τα υπόλοιπα σημαίνουν ότι στη θύρα 8160, τα δεδομένα θα αποστέλλονται μέσω http χρησιμοποιώντας ως μορφή κωδικοποίησης του βίντεο το h264 ως την προεπιλεγμένη ροή εξόδου.

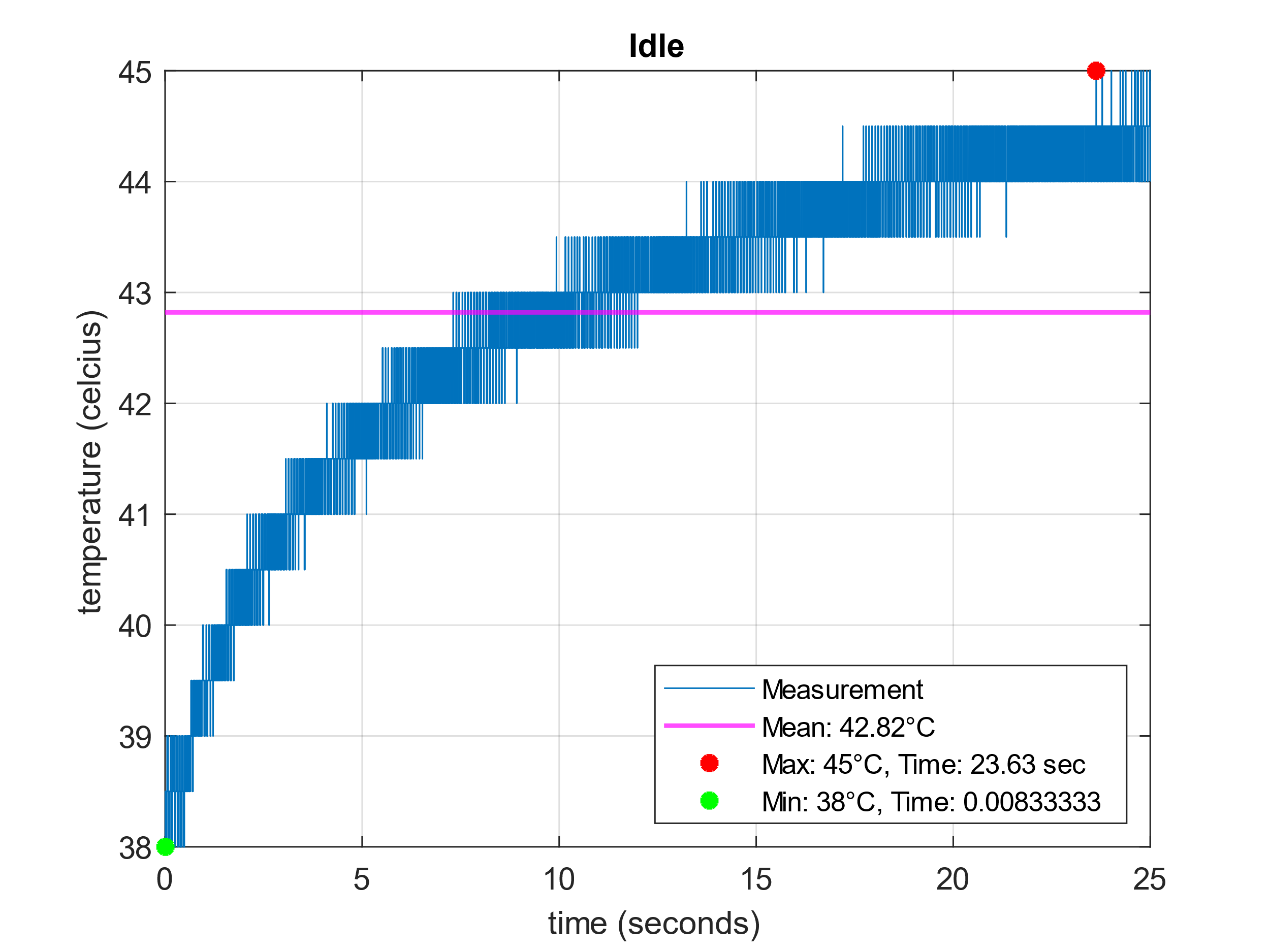
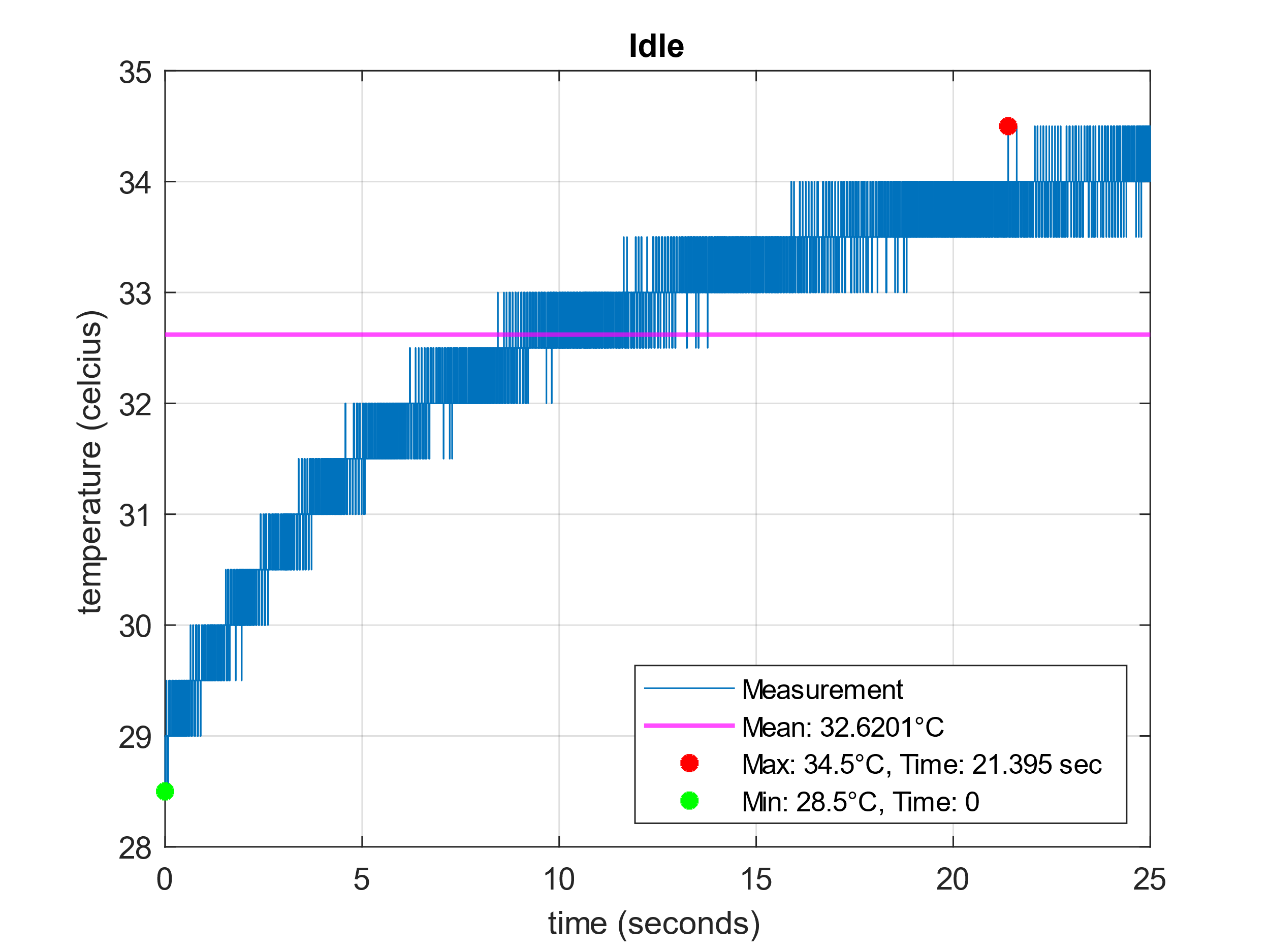
Η περίοδος καταγραφής όλων των μετρήσεων είναι 25 λεπτά.

###### Without Movement:



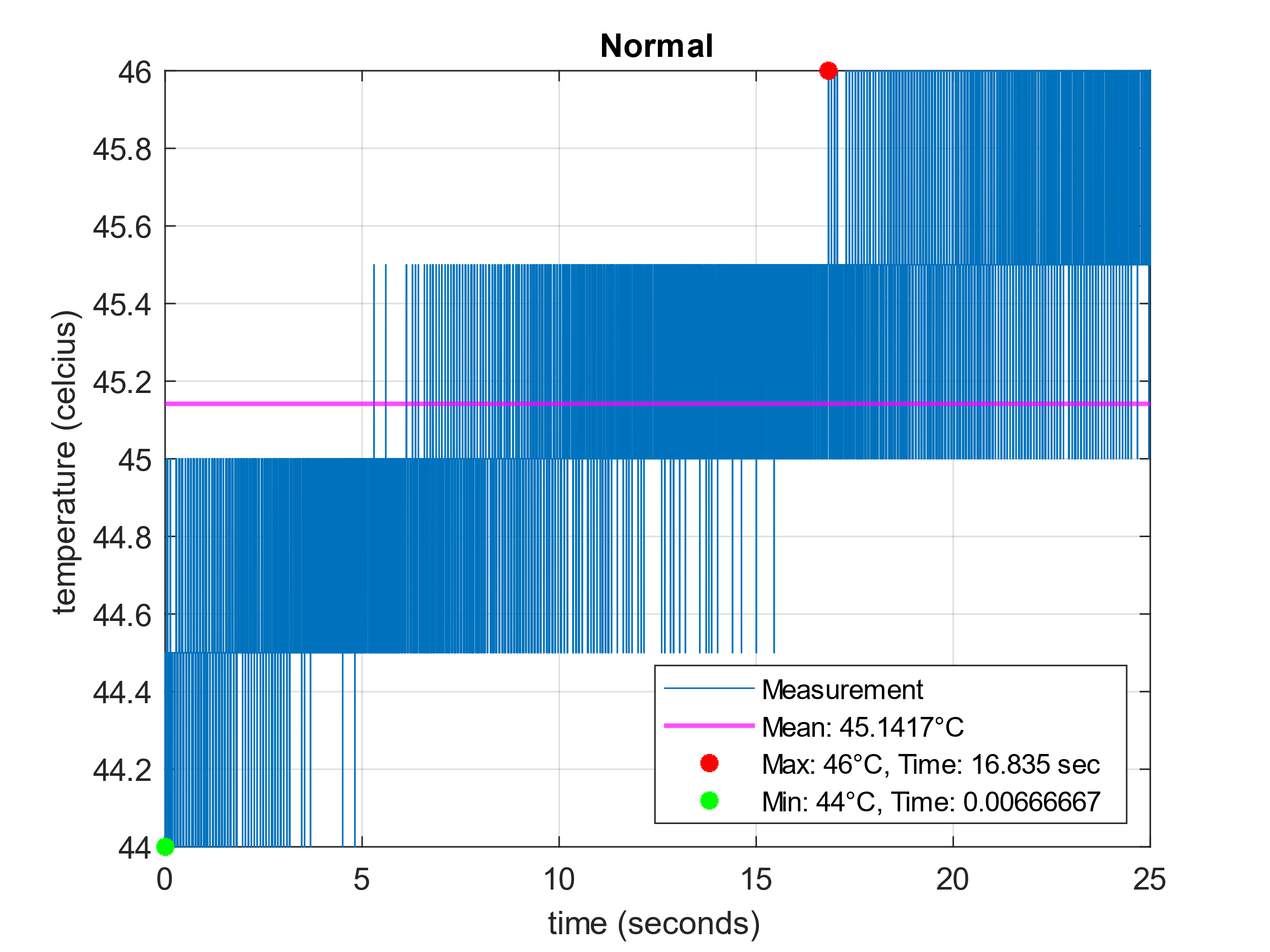
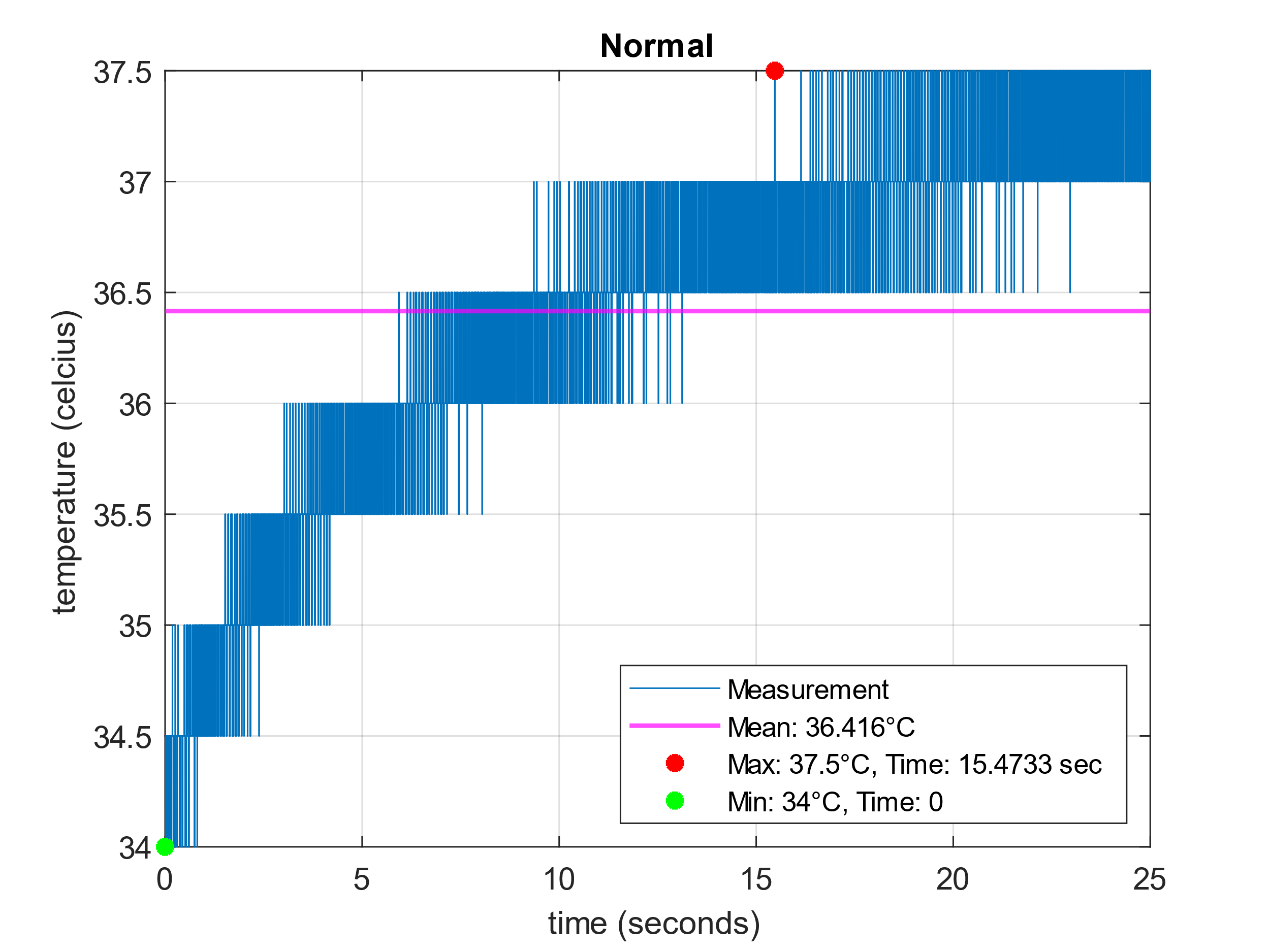
Εικόνα 2: Boot without attack Εικόνα 3: Boot with attack

Η καταγραφή κατά την εκκίνηση του συστήματος ξεκίνησε κατευθείαν μόλις συνδέθηκε το Raspberry Pi στον υπολογιστή μέσω της θύρας USB. Επίσης, καθ’ όλη την διάρκεια εκκίνησης δεν εκτελούταν κανένα πρόγραμμα πέραν των βασικών του λειτουργικού συστήματος του Raspberry Pi.



Εικόνα 4: Idle without attack Εικόνα 5: Idle with attack

Ως κατάσταση αδράνειας του συστήματος θεωρήθηκε η συνεχής ροή εικόνας από την κάμερα χωρίς κίνηση.

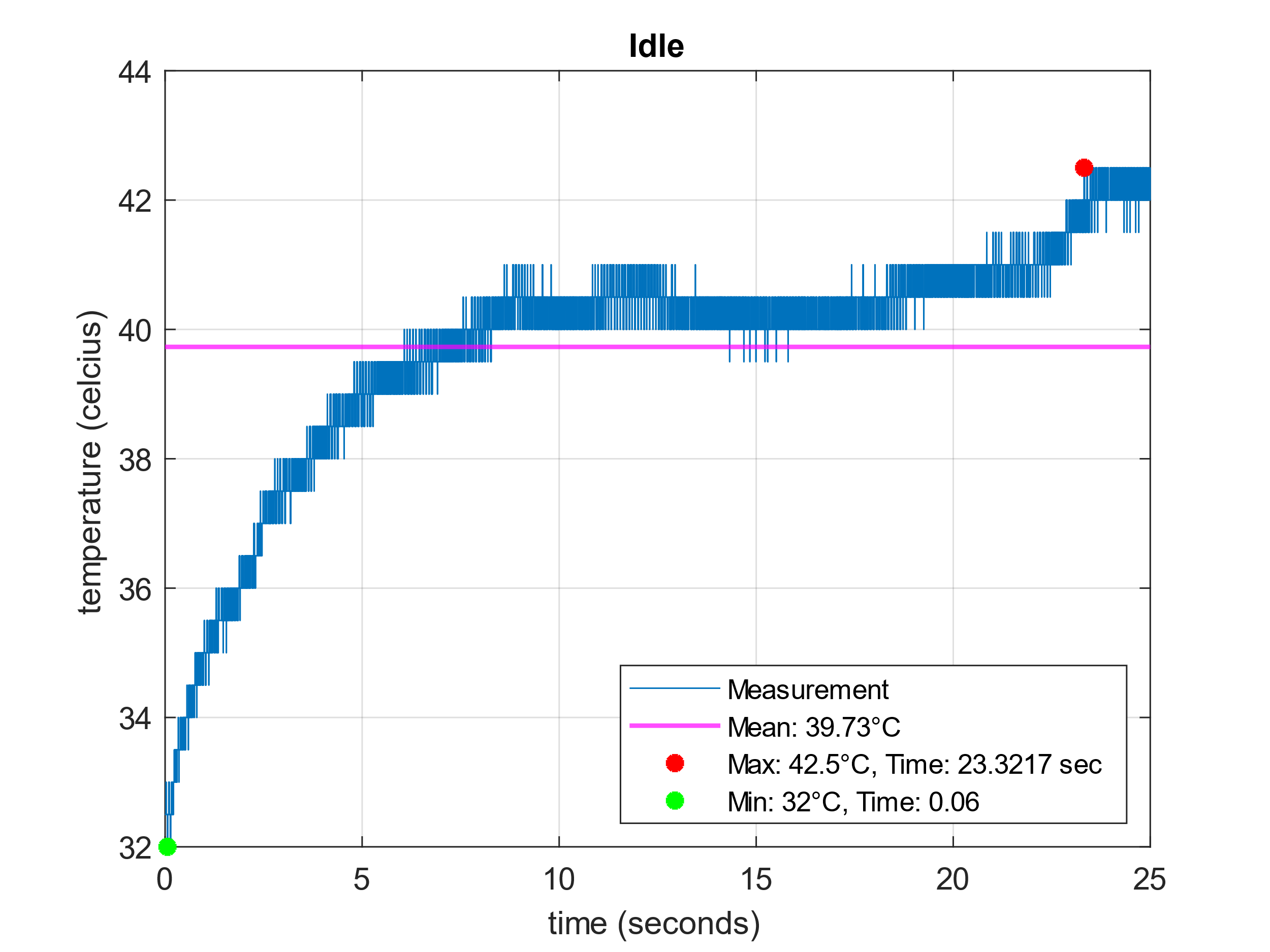


Εικόνα 6: Normal without attack Εικόνα 7: Normal with attack

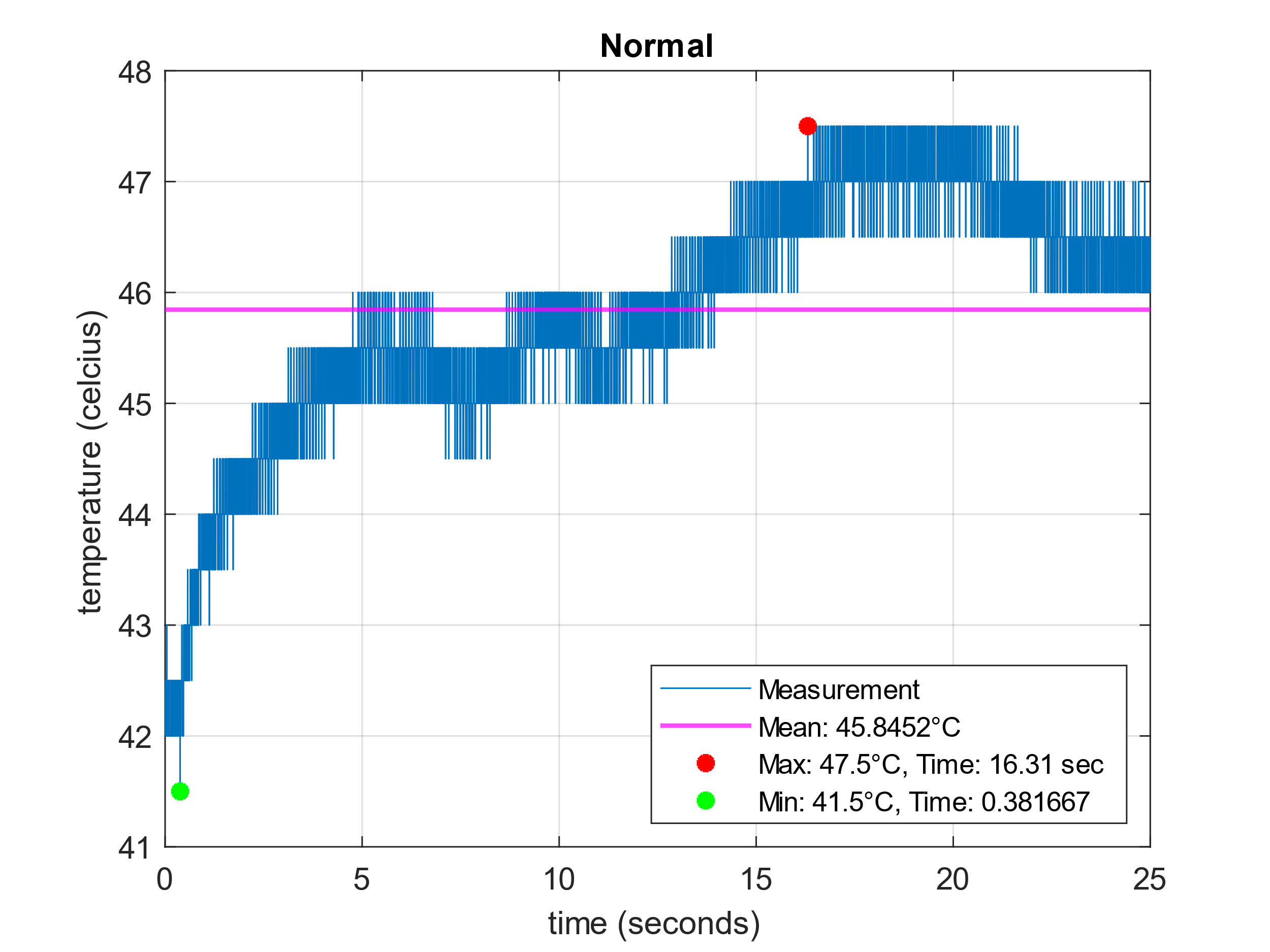
Ως η κανονική λειτουργία του συστήματος θεωρήθηκε η συνεχής ροή εικόνας από την κάμερα χωρίς κίνηση και παράλληλα η εκτέλεση ενός προγράμματος σε γλώσσα προγραμματισμού Python, που μας επιστρέφει την εσωτερική θερμοκρασία του συστήματος.

###### With Movement:

Το διάγραμμα κατά την διάρκεια της εκκίνησης παραλήφθηκε διότι οι συνθήκες ήταν οι ίδιες με αυτές του προηγούμενου, επηρεάζονται μόνο η κατάσταση αδράνειας και η κανονική λειτουργία του συστήματος. Στη συγκεκριμένη καταγραφή προστέθηκε κίνηση κατά τη συνεχή ροή εικόνας.

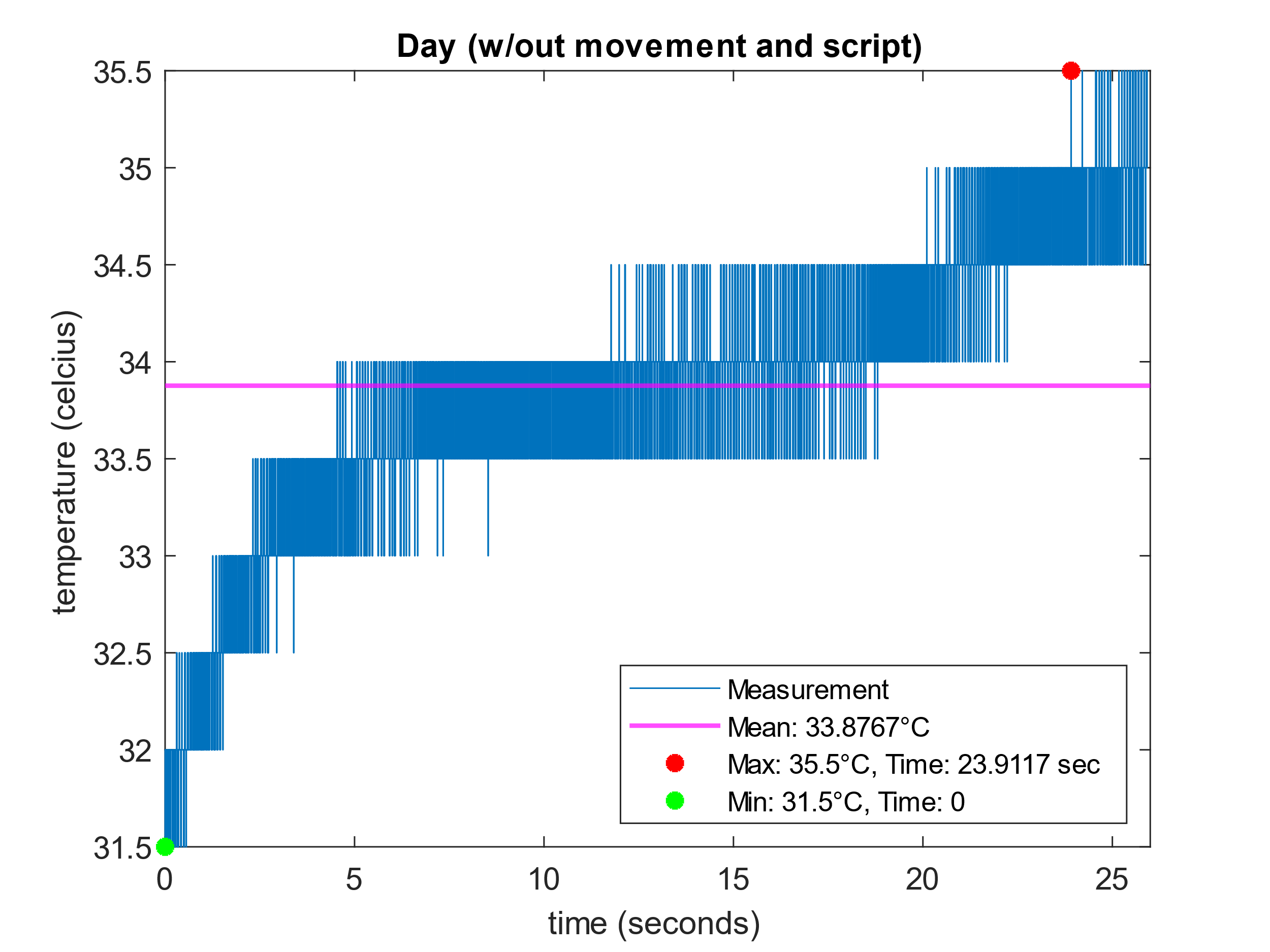


Εικόνα 8: Idle without attack



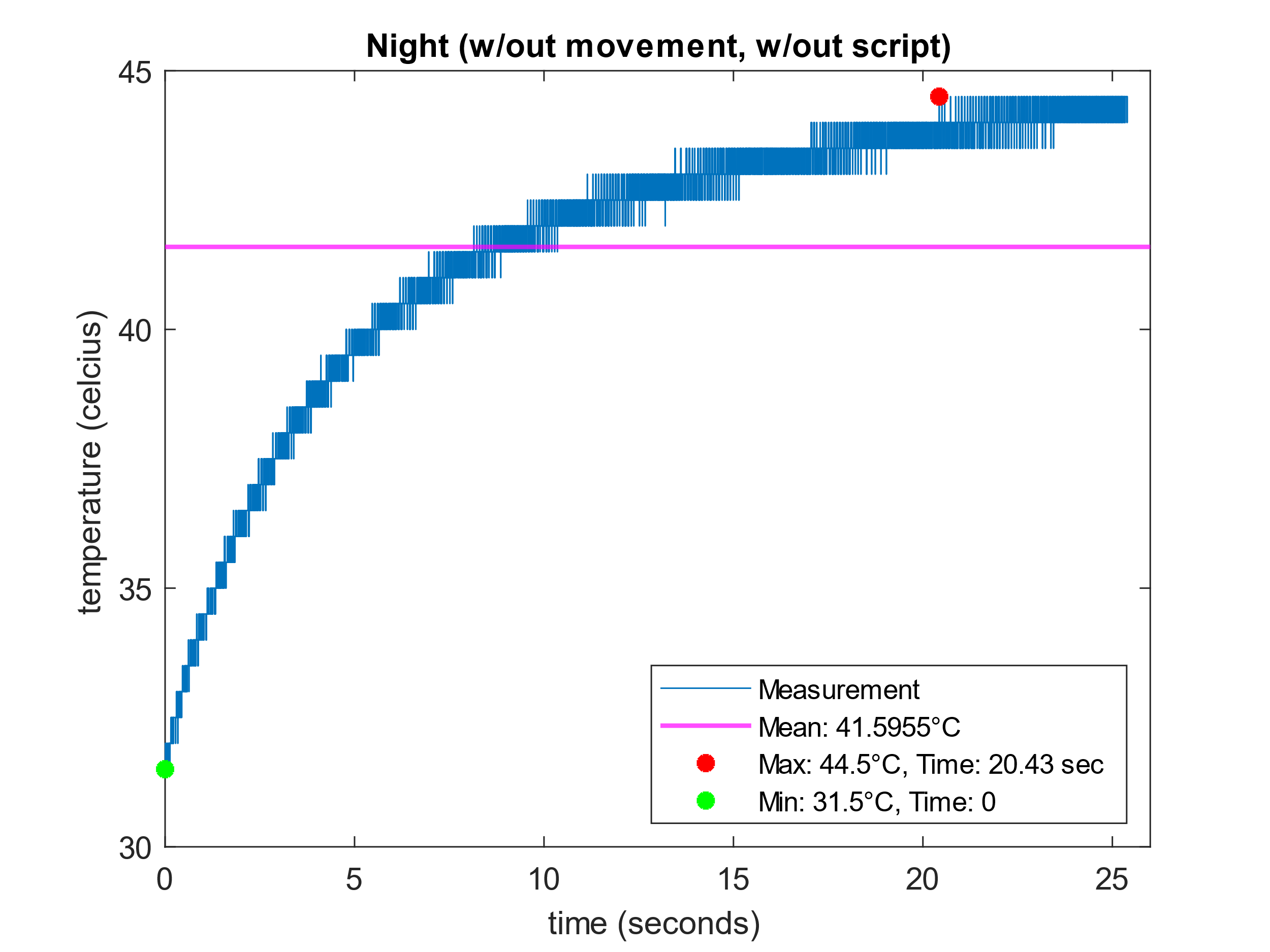
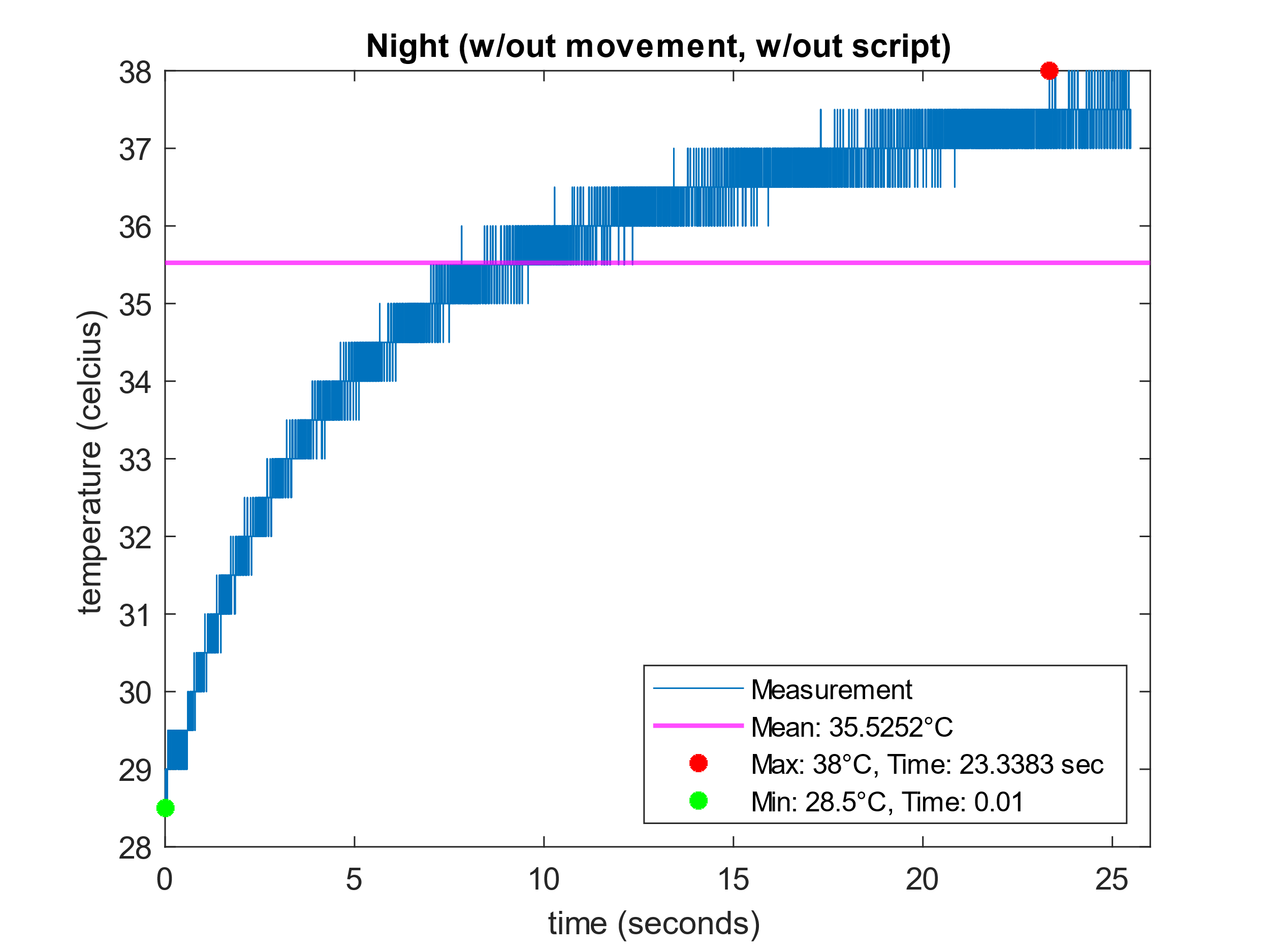
Εικόνα 9: Normal without attack

###### Without Movement / Without script:



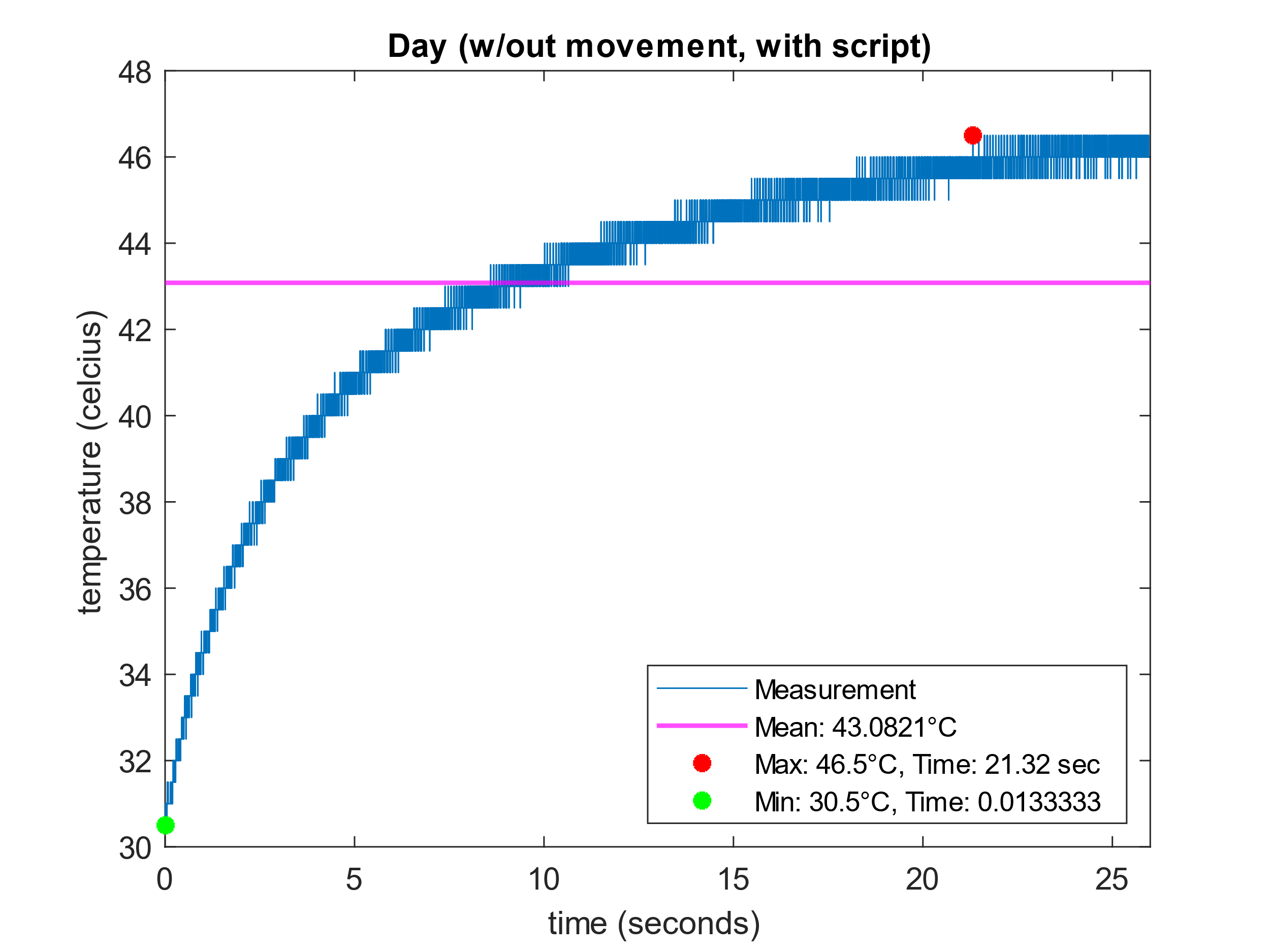
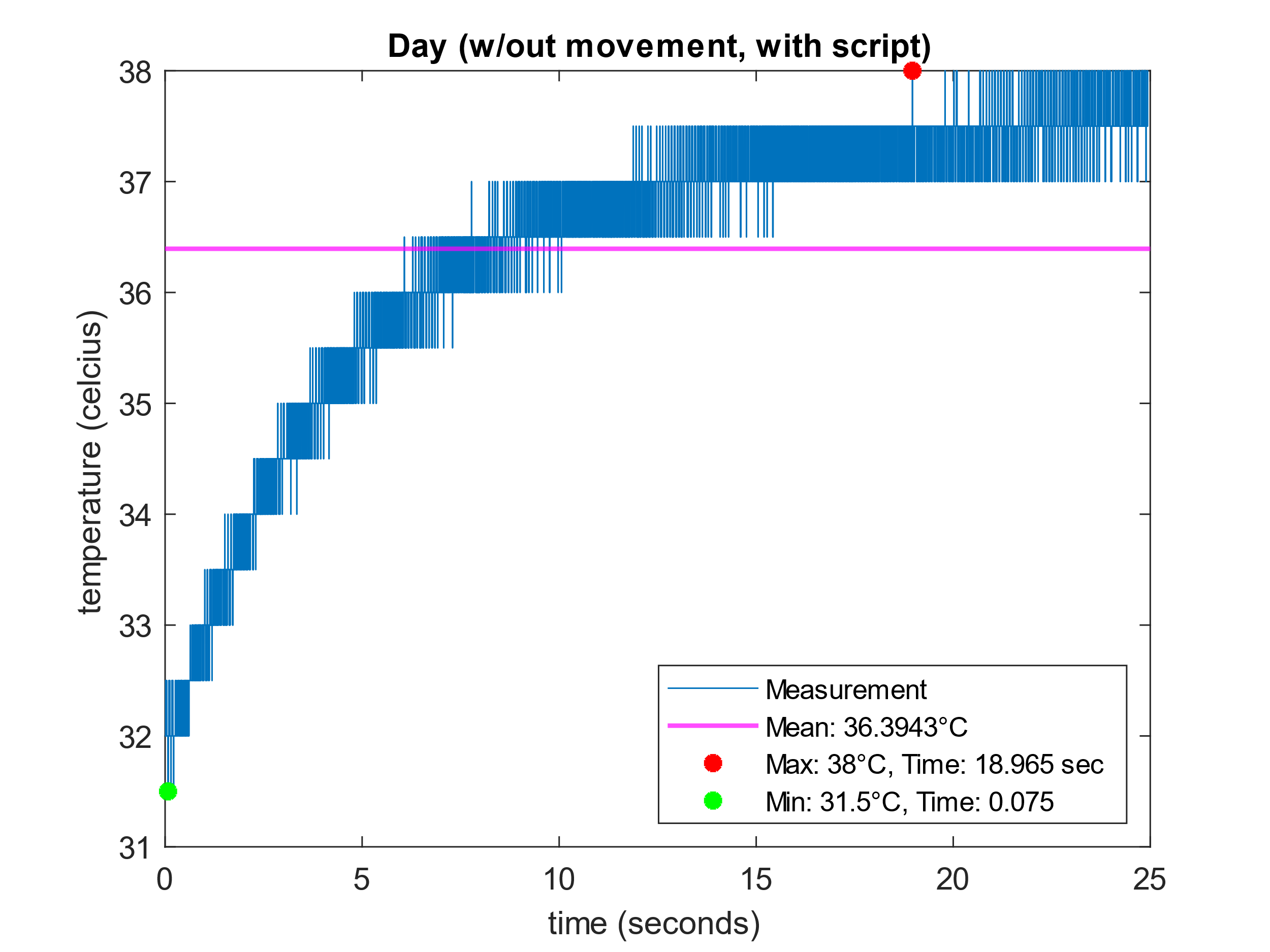
Εικόνα 10: Without Attack Εικόνα 11: With Attack

Στην Εικόνα 10 παρατηρούμε την καταγραφή της θερμοκρασίας του επεξεργαστή της κάμερας σε ημερήσια λήψη χωρίς να εκτελείται κάποια επίθεση. Το σενάριο λειτουργίας της κάμερας αφορά την περίπτωση χωρίς κίνηση κατά την καταγραφή, και χωρίς την εκτέλεση κάποιου σεναρίου λογισμικού στο παρασκήνιο. Κατά την μέτρηση αυτή, παρατηρείται ελάχιστη τιμή θερμοκρασίας στους 31.5 οC, μέγιστη στους 36.5 οC

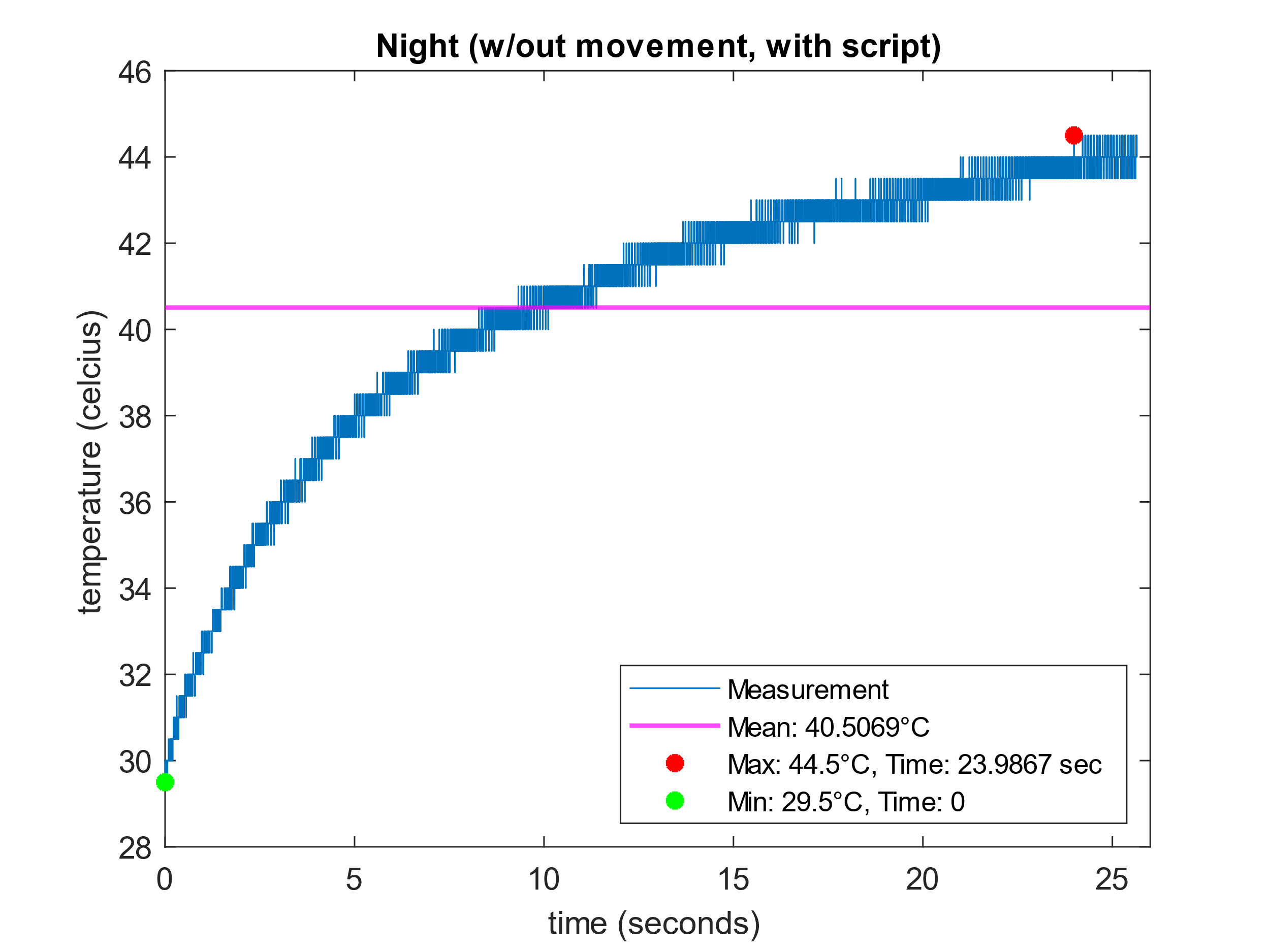
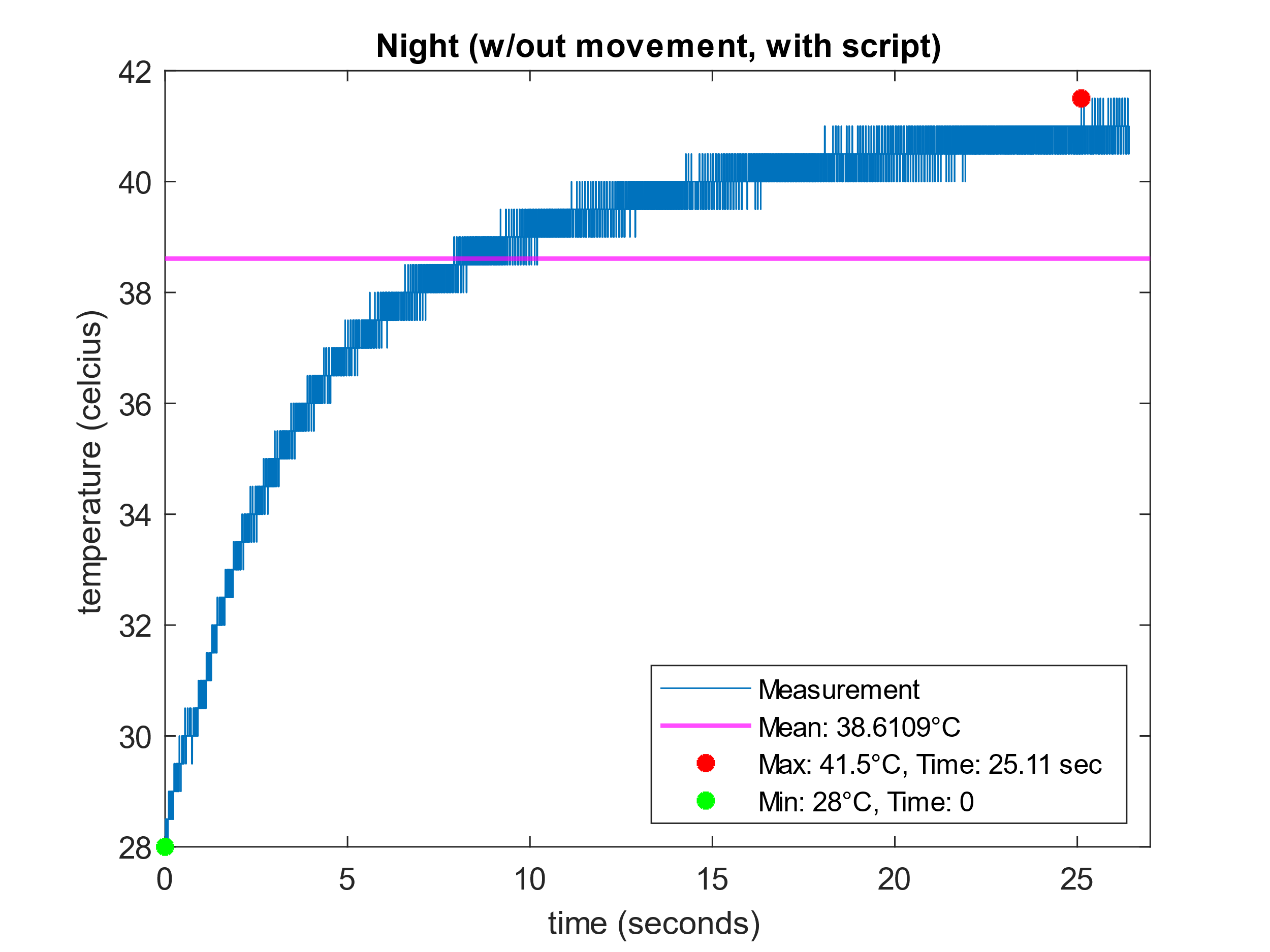


Εικόνα 12: Without Attack Εικόνα 13: With Attack

###### Without Movement / With Script:

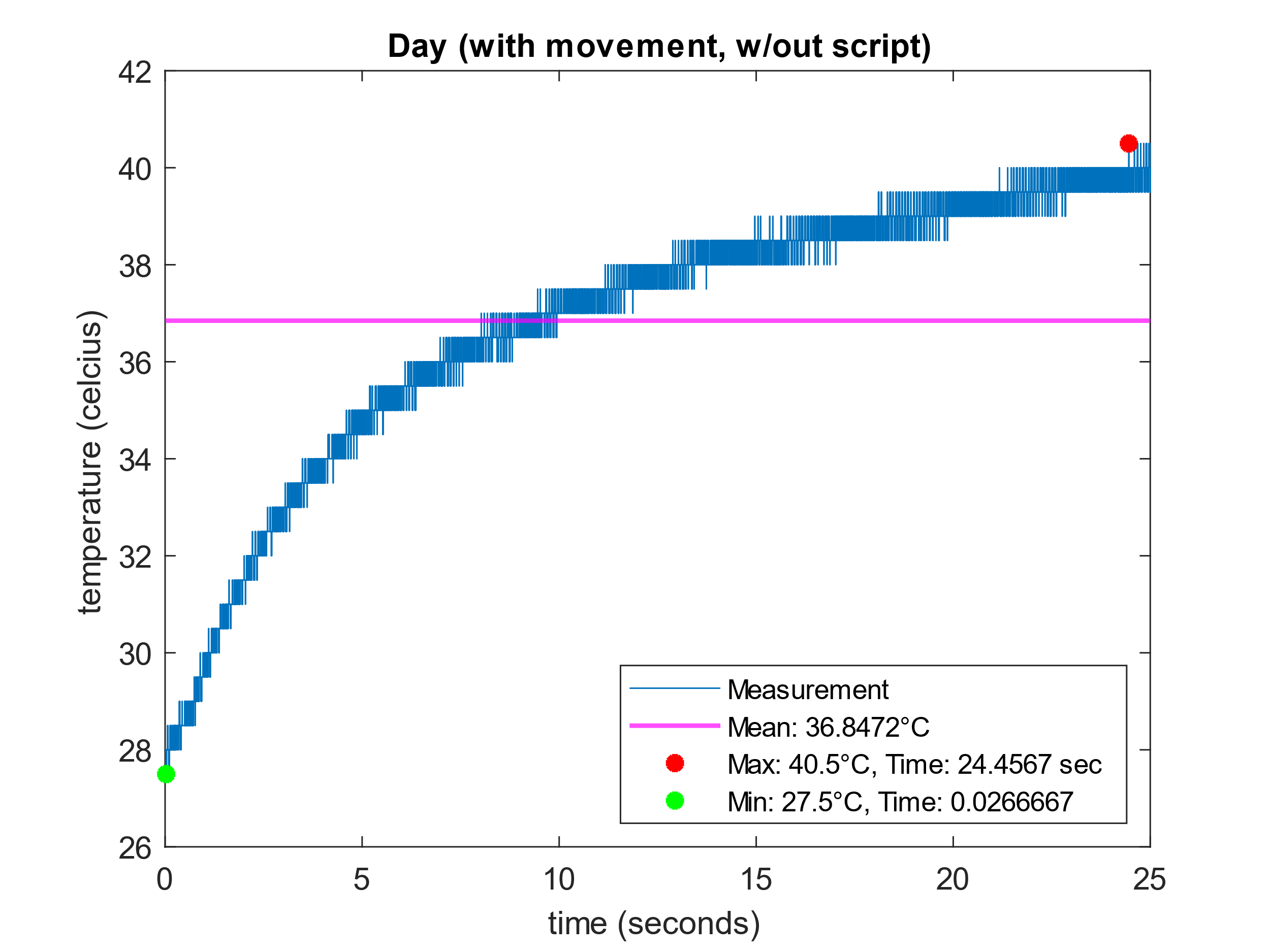
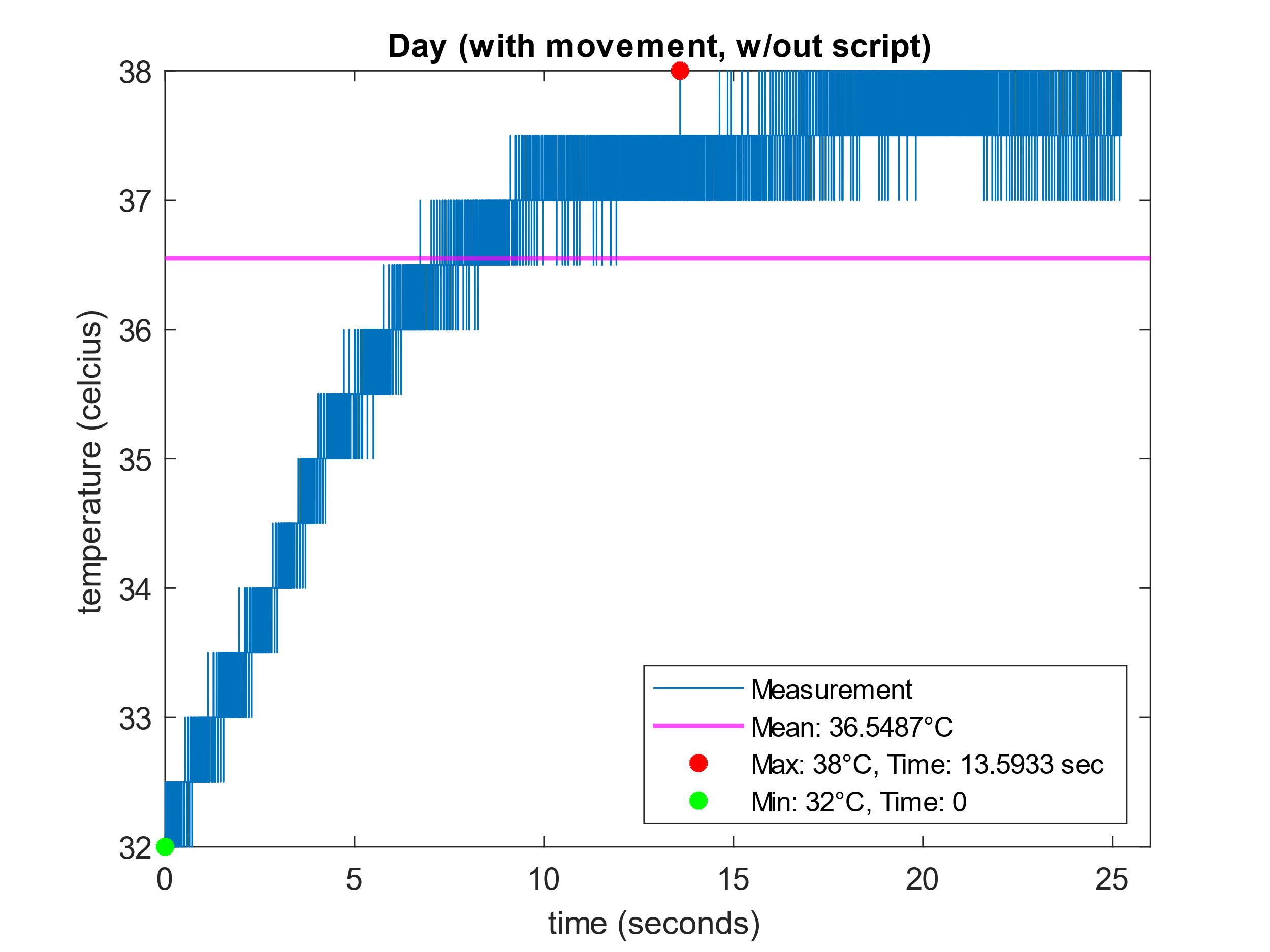


Εικόνα 14: Without Attack Εικόνα 15: With Attack

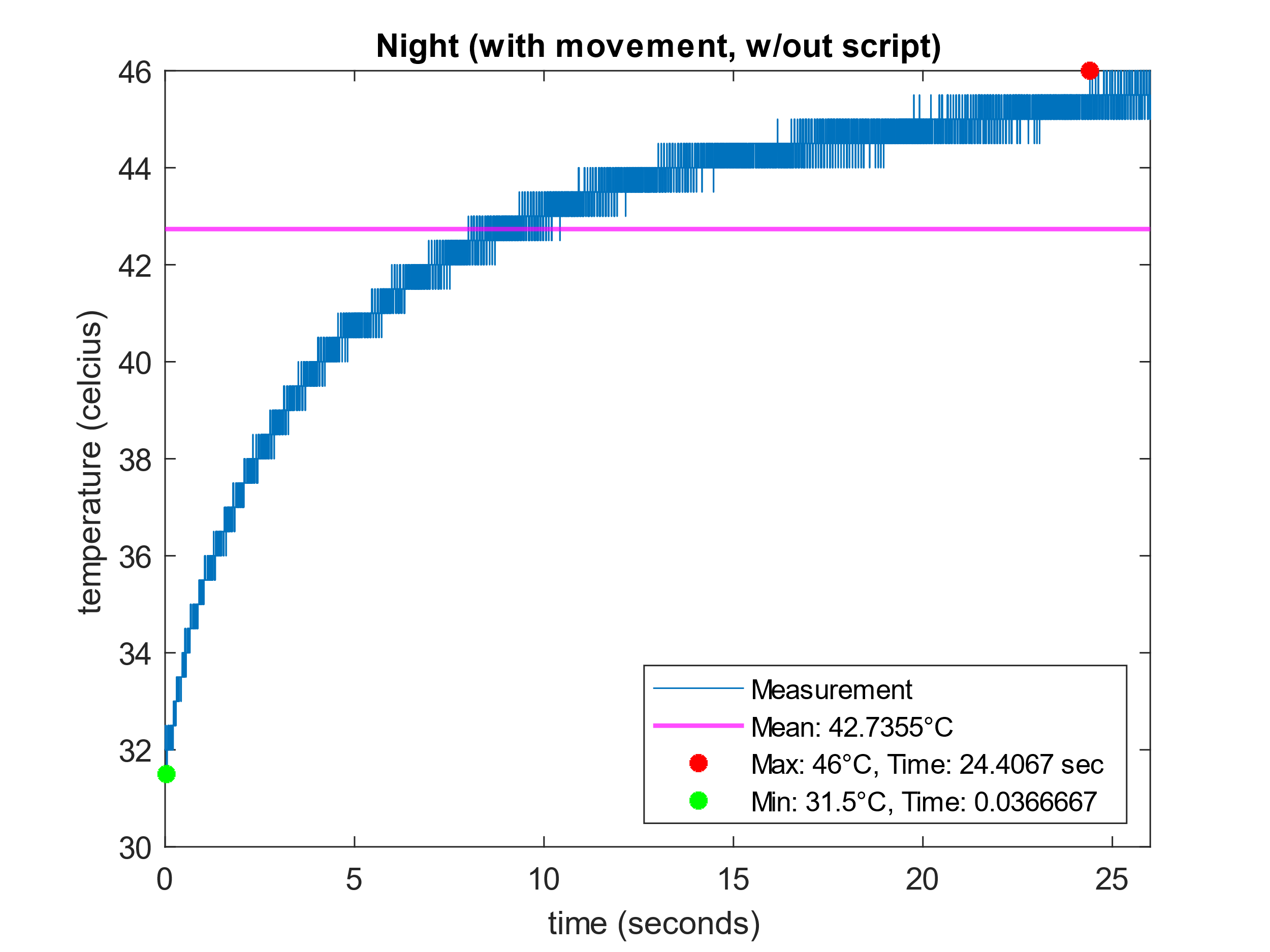
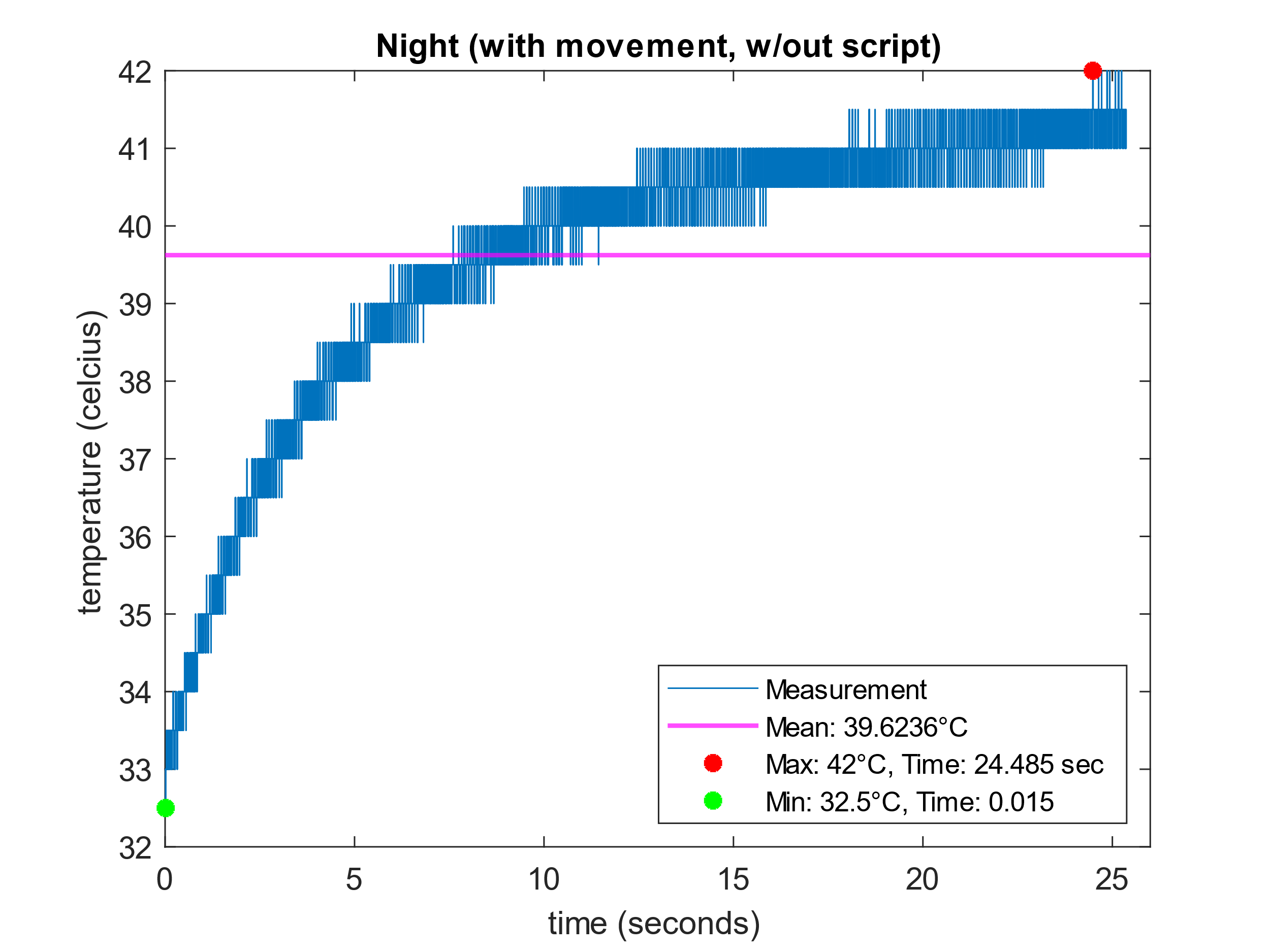


Εικόνα 16: Without Attack Εικόνα 17: With Attack

###### With Movement / Without Script:

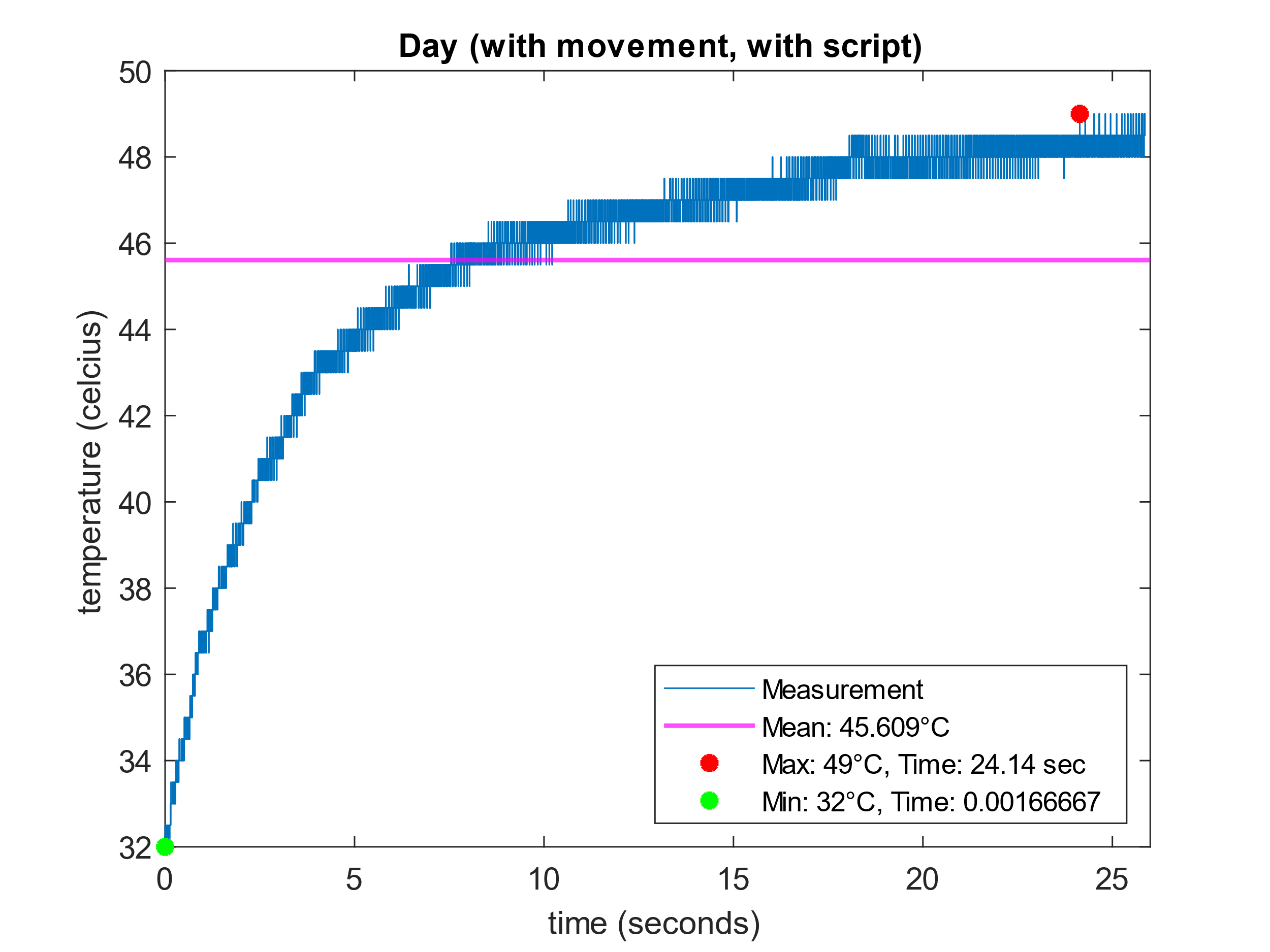
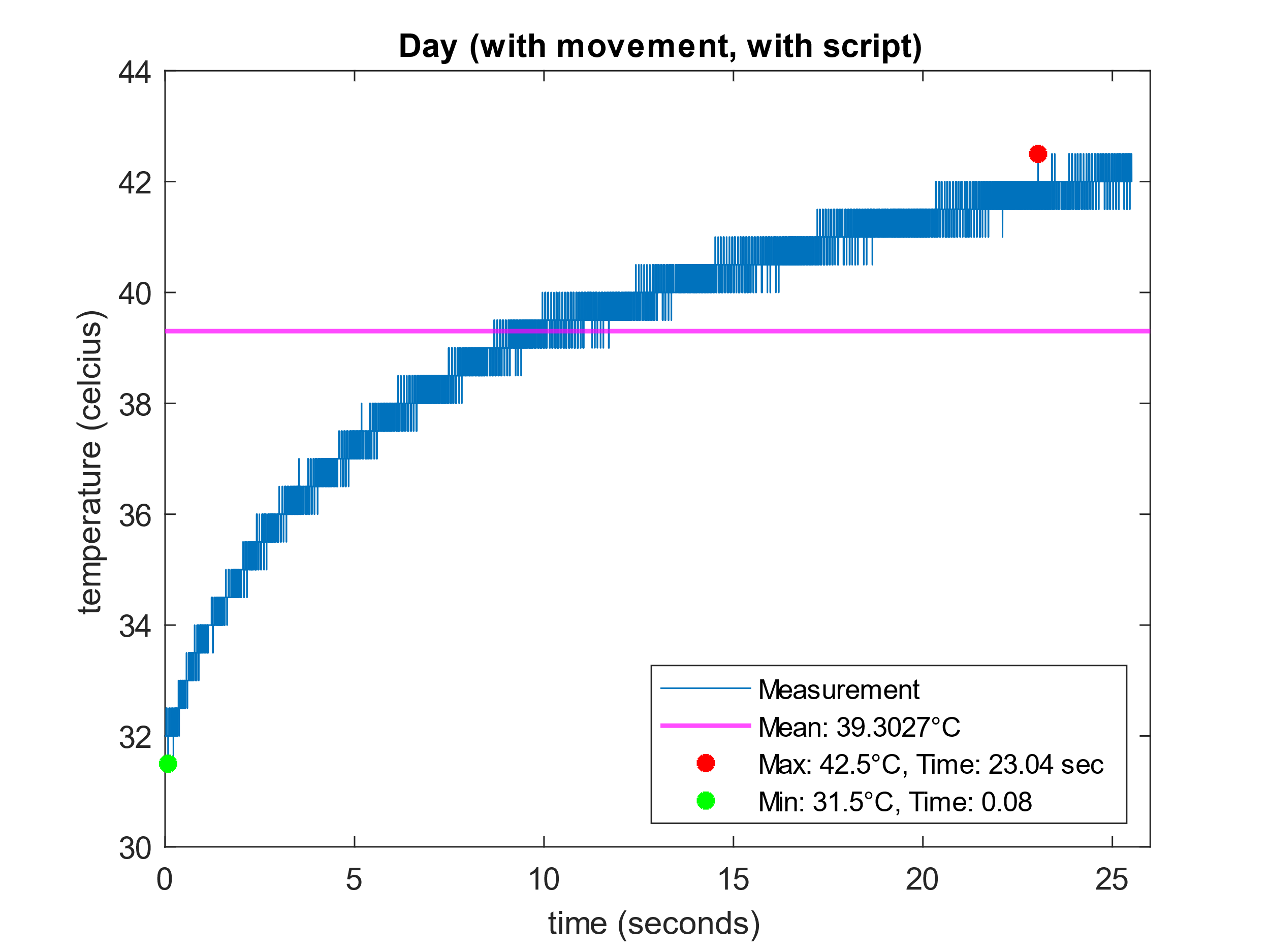


Εικόνα 18: Without Attack Εικόνα 19: With Attack

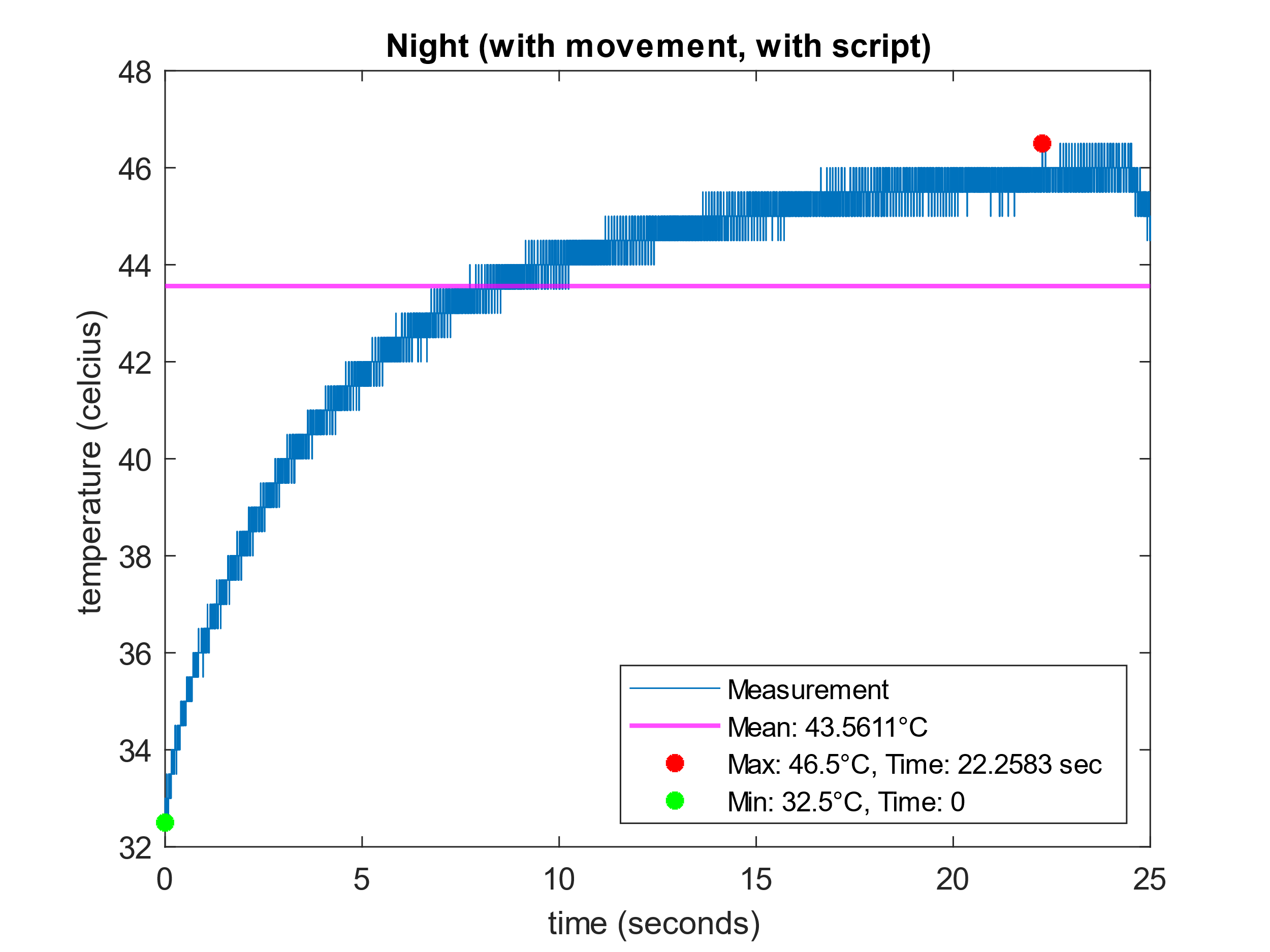
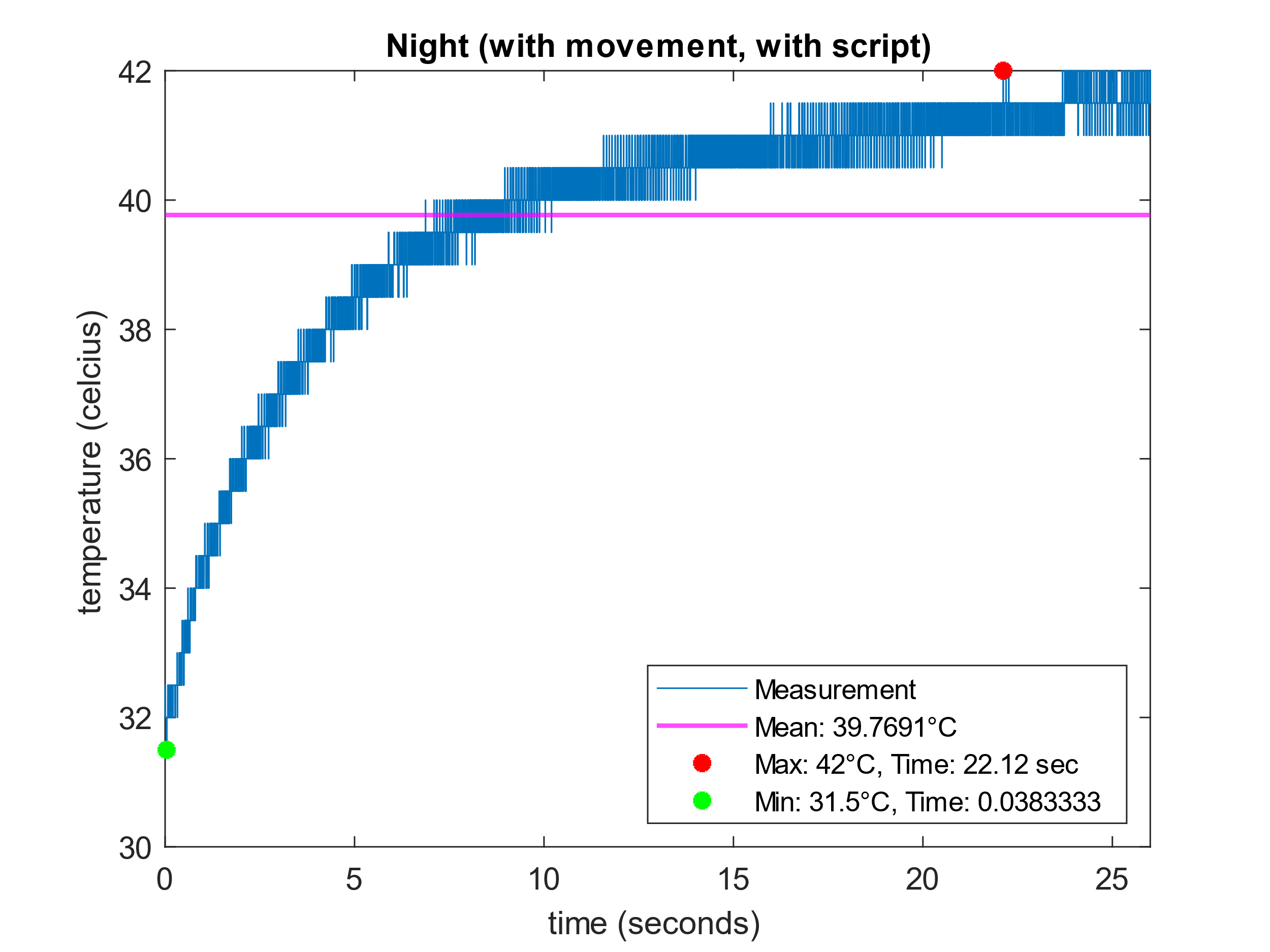


Εικόνα 20: Without Attack Εικόνα 21: With Attack

###### With Movement / With Script:



Εικόνα 22: Without Attack Εικόνα 23: With Attack



Εικόνα 24: Without Attack Εικόνα 25: With Attack