

Σχετικά με το Project :

Το BabySpy είναι μια έξυπνη συσκευή παρακολούθησης παιδικού δωματίου. Σκοπός της συσκευής είναι με χρήση αισθητήρων να καταγράφει μετρήσεις και ειδοποιείται ο γονέας.

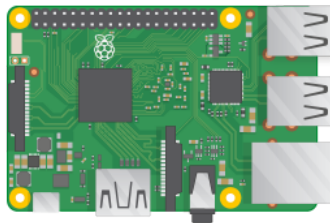
Βασικές λειτουργίες της συσκευής :

- Καταγραφή της θερμοκρασίας του χώρου
- Καταγραφή της υγρασίας του χώρου
- Δυνατότητα Live Streaming απευθείας με την συσκευή
- Δυνατότητα απομακρυσμένης αναπαραγωγής νανουρίσματος
- Δυνατότητα απομακρυσμένης ενεργοποίησης λάμπας φωτός

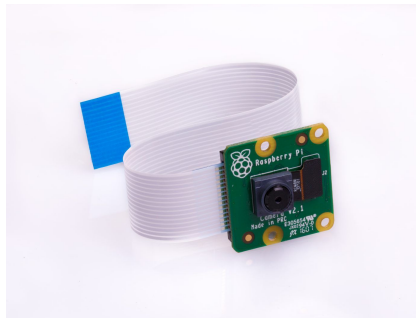
Ο γονιός θα έχει επιπλέον την δυνατότητα να δει τα καταγεγραμμένα δεδομένα της συσκευής σε μια online πλατφόρμα.

Για την υλοποίηση του Project θα χρειαστούμε τα παρακάτω υλικά :

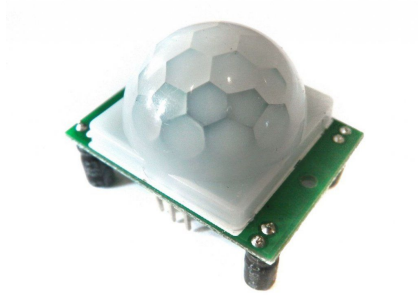
1) Raspberry PI 3 B+



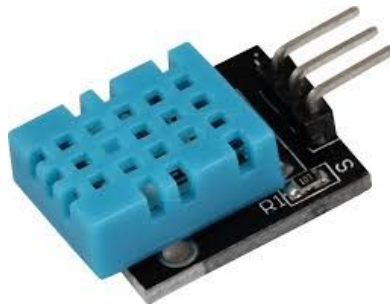
2) Pi Camera



3) Αισθητήρα ανίχνευσης κίνησης (PIR)



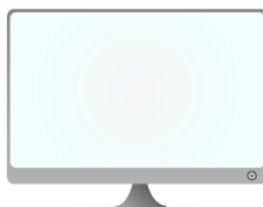
4) Αισθητήρα θερμοκρασίας και υγρασίας(DHT11)



5) Τροφοδοτικό για το Raspberry PI



6) Οθόνη με σύνδεση HDMI



7) Καλώδιο HDMI



8) Πληκτρολόγιο με USB



9) Ποντίκι με USB



10) Μνήμη SD



11) 6 θηλυκά - θηλυκά Καλώδια



Κοστολόγηση Προϊόντων

Raspberry PI 3 B+	38.70
Pi Camera	29.40
Pir Sensor	2.80
DHT11 Sensor	2.20
Τροφοδοτικό Raspberry	9.90
Οθόνη	-----
HDMI	1.02
<u>Πληκτρολόγιο Usb</u>	<u>3.00</u>
<u>Ποντίκι Usb</u>	<u>2.17</u>
<u>Μνήμη SD 8GB</u>	<u>4.43</u>
<u>female cable</u>	

Τι Λογισμικό χρειαζόμαστε για το Raspberry Pi:

Για την λειτουργία του Raspberry Pi Θα χρειαστούμε να εγκαταστήσουμε το Λειτουργικό Σύστημα NOOBS στην μνήμη SD

Βήματα εγκατάστασης Λ.Σ. NOOBS

1ο βήμα

Το πρώτο που χρειάζεται είναι να κάνουμε είναι να συνδέσουμε την κάρτα Micro SD με τον ανάπτορα στον υπολογιστή μας, και να τη διαμορφώσουμε κατάλληλα. Το ίδιο το NOOBS συνιστά να χρησιμοποιήσουμε την επίσημη εφαρμογή για Format σε SD, την οποία θα βρούμε στη διεύθυνση:

https://www.sdcard.org/downloads/formatter_4/eula_windows/

Κατεβάζουμε την εφαρμογή κάνοντας κλικ στο Accept στο μέρος της σελίδας.

2ο βήμα

Αποσυμπιέζουμε το .zip, εγκαθιστούμε και τρέχουμε την εφαρμογή.

3ο βήμα

Στο παράθυρο της εφαρμογής επιλέγουμε το “Option”, στο οποίο αλλάζουμε το Format Size Adjustment σε “ON”.

4ο βήμα

Κάνουμε κλικ στο Format, και σε λίγα δευτερόλεπτα η διαδικασία διαμόρφωσης έχει ολοκληρωθεί.

5ο βήμα

Με την κάρτα SD μας διαμορφωμένη, μπαίνουμε στη διεύθυνση <https://www.raspberrypi.org/downloads/noobs/> και κατεβάζουμε την τελευταία έκδοση του NOOBS, είτε απευθείας από την ιστοσελίδα με μορφή συμπιεσμένου φακέλου .zip, είτε μέσω Torrent. Το πλήρες Noobs έχει ενσωματωμένο το λειτουργικό σύστημα Raspbian, που είναι το πιο δημοφιλές λειτουργικό σύστημα για την εγκατάσταση Raspberry Pi, και μας δίνει πλήθος δυνατοτήτων.

6ο βήμα

Αφού κατεβάσουμε το συμπιεσμένο φάκελο .zip, κάνουμε εξαγωγή όλων των περιεχομένων του.

7ο βήμα

Ανοίγουμε το φάκελο στον οποίο έγιναν αποσυμπίεση και αντιγράφουμε όλα τα αρχεία στην κάρτα SD.

8ο βήμα

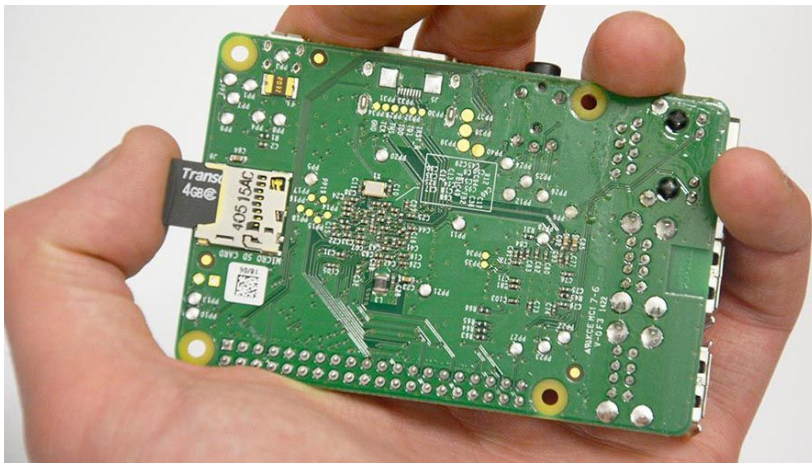
Αφού ολοκληρωθεί η αντιγραφή, κάνουμε ασφαλή αφαίρεση της κάρτας SD, για να ελαχιστοποιήσουμε την πιθανότητα καταστροφής δεδομένων.

Εγκατάσταση Raspberry Pi με το Raspbian

Εγκατάσταση μνήμης SD

Γυρίζουμε ανάποδα το Raspberry Pi για να τοποθετήσουμε την Micro SD. Υπάρχει ένας μόνο τρόπος για να μπει, και μπαίνοντας θα “κλειδώσει”.

Για να αφαιρέσουμε την SD, την ξαναπατάμε προς τα μέσα, για να ξεκλειδώσει.



Σύνδεση Καλωδίων στο Raspberry Pi

ΠΡΟΣΟΧΗ : Δεν συνδέουμε ακόμα την τροφοδοσία ρεύματος γιατί το Raspberry δεν έχει κουμπί on/off με αποτέλεσμα την εκκίνηση του.

Συνδέουμε όλα τα απαιτούμενα καλώδια στο Raspberry όπως:

- 1)Καλωδιο HDMI η άλλη άκρη στην οθόνη
- 2)Πληκτρολόγιο
- 3)Ποντίκι

Πρώτη εκκίνηση του Raspberry Pi

Έχοντας συνδέσει όλα τα καλώδια και της τροφοδοσίας και έχουμε εγκαταστήσει και την μνήμη SD,είμαστε έτοιμοι για την πρώτη εκκίνηση του Raspberry.

Ξεκινώντας, θα μας δείξει μια οθόνη με διάφορα χρώματα, γνωστή και σαν rainbow screen.Σύντομα θα φορτώσει το NOOBS. Αν το Raspberry Pi δεν είναι συνδεδεμένο στο Internet, θα εμφανίσει στη λίστα μόνο το Raspbian. Αν το Raspberry Pi είναι συνδεδεμένο στο Internet, το NOOBS θα μας δείξει και εναλλακτικά λειτουργικά συστήματα που μπορούμε να επιλέξουμε. Θα πρέπει όμως να περιμένουμε να κατεβάσει το καθένα από αυτά από το Internet.Το πρώτο βήμα είναι στο κάτω μέρος της οθόνης να ορίσουμε σαν γλώσσα English (US) και σαν πληκτρολόγιο us. Τα ελληνικά δεν υπάρχουν σαν επιλογή, θα τα ρυθμίσουμε μετά την εγκατάσταση.Ο λόγος είναι πως το προεπιλεγμένο πληκτρολόγιο του Ηνωμένου Βασιλείου έχει ελαφρώς διαφορετική διάταξη από τα πληκτρολόγια που έχουμε στην Ελλάδα, κυρίως όσον αφορά κάποια σύμβολα όπως το @ και το #.Έχοντας τσεκάρει το Raspbian και κάνοντας κλικ στο Install, το σύστημα μας προειδοποιεί πως θα διαγραφεί όλο το περιεχόμενο της SD.Επιλέγοντας “Yes”, ξεκινάει η εγκατάσταση. Ανάλογα με την ταχύτητα της κάρτας SD που έχουμε βάλει, θα πάρει αρκετή ώρα.Εφόσον όλα πάνε καλά, το σύστημα θα μας εμφανίσει το μήνυμα πως το λειτουργικό σύστημα (ή τα λειτουργικά συστήματα, αν

επιλέξαμε πολλαπλά) εγκαταστάθηκαν επιτυχώς.Κάνοντας κλικ στο OK, το Raspberry Pi θα κάνει επανεκκίνηση. Στην επόμενη εκκίνηση, θα μας βάλει στο περιβάλλον του Raspbian.

Στη συνέχεια ενημερώνουμε (update) τα πακέτα του συστήματος πληκτρολογώντας στο LXTerminal ή από την γραμμή εντολών:

```
sudo apt-get update
```

Μετά αναβαθμίζουμε (upgrade) τα εγκατεστημένα πακέτα με την εντολή

```
sudo apt-get dist-upgrade
```

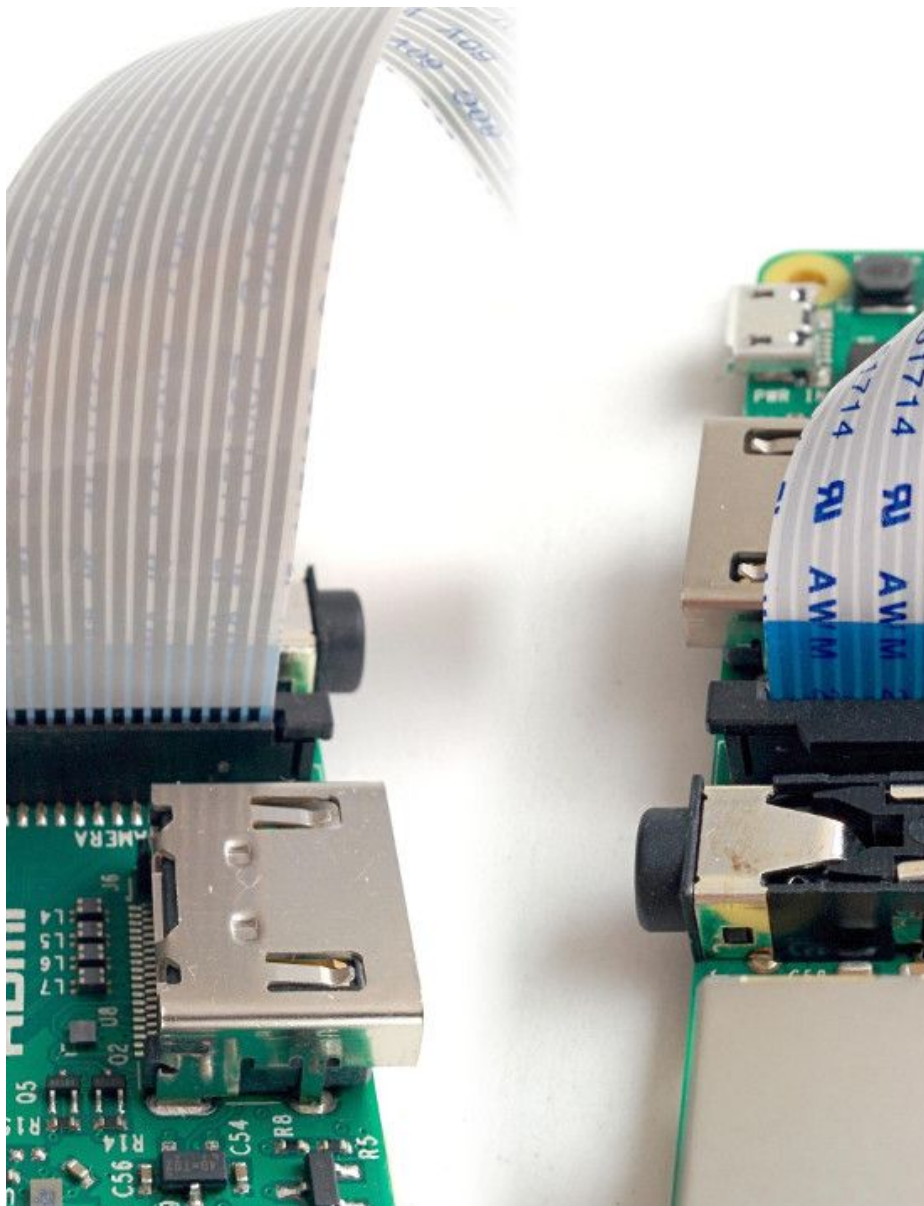
Μόλις τελειώσατε με την εγκατάσταση και ενημέρωση του NOOBS.

Προετοιμασία της Pi Κάμερας

Συνδεσμολογία Κάμερας

ΠΡΟΣΟΧΗ:

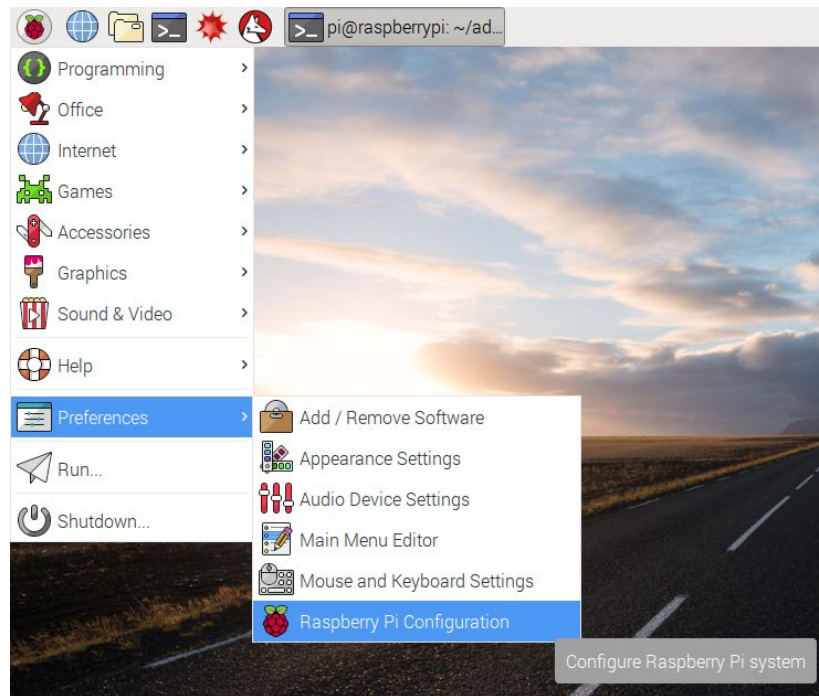
Το καλώδιο με τα γράμματα να κοιτάει προς την μεριά των
USB



Παραμετροποίηση Raspberry Pi

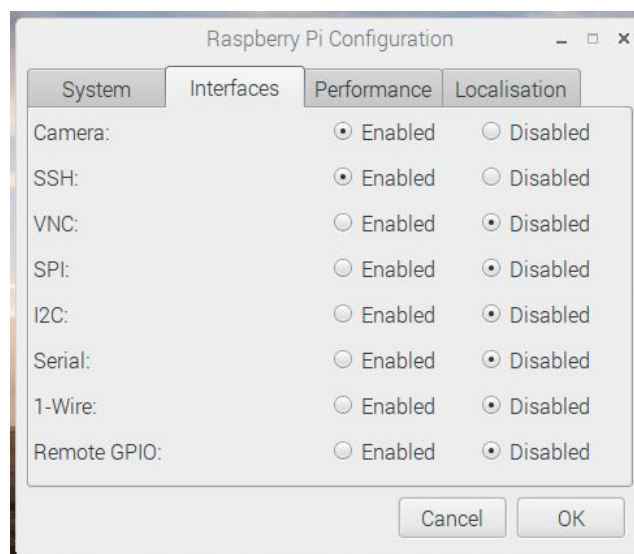
1ο Βήμα

Ανοίγουμε το Raspberry Pi Configuration Tool από τις προτιμήσεις στο κυρίως μενού:



2ο Βήμα

Επιβεβαιώστε ότι το λογισμικό της κάμερας είναι ενεργοποιημένο:



Προγραμματισμός του Raspberry

Πριν Ξεκινήσουμε να γράφουμε τον κώδικα μας , είναι απαραίτητο να εγκαταστήσουμε τις απαραίτητες Βιβλιοθήκες (Libraries).

Για το προγραμματισμό της κάμερας θα χρειαστούμε τις παρακάτω βιβλιοθήκες :

- pi camera | Website:
<https://picamera.readthedocs.io/en/release-1.13/install.html>
- io | Προεγκατεστημένη
- logging | Προεγκατεστημένη
- socketserver | Προεγκατεστημένη
- threading | Προεγκατεστημένη
- http | Προεγκατεστημένη

Εγκατάσταση της βιβλιοθήκης pi camera

- 1) Ανοίγουμε το τερματικό
- 2) Ενημερώνουμε τη λίστα αποθετηρίων σας:

```
sudo apt-get update
```

- 3) Στη συνέχεια, εγκαθιστούμε το πακέτο για το Python 3:

```
sudo apt-get install python-picamera python3-picamera
```

Προγραμματισμός της κάμερας

Προγραμματισμός της κάμερας για web streaming στο φυλλομετρητή

1ο Βήμα

Πηγαίνουμε στην ιστοσελίδα της Raspberry Pi κάμερας

<https://picamera.readthedocs.io/en/release-1.13/index.html>

2ο Βήμα

Πηγαίνουμε στην παράγραφο 4 (Advanced Recipes)

The screenshot shows the Picamera documentation website. On the left is a dark sidebar with a list of navigation items: 1. Installation, 2. Getting Started, 3. Basic Recipes, 4. Advanced Recipes (highlighted with a white box), 5. Frequently Asked Questions (FAQ), 6. Camera Hardware, 7. Development, 8. Deprecated Functionality, 9. API - The PiCamera Class, 10. API - Streams, 11. API - Renderers, 12. API - Encoders, 13. API - Exceptions, 14. API - Colors and Color Matching, 15. API - Arrays, 16. API - mmalobj, 17. Change log, and 18. License. At the bottom of the sidebar are links for 'Read the Docs' and 'v: release-1.13'. The main content area has a header with 'Docs » 1. Installation' and a link to 'Edit on GitHub'. Below the header, the section '1. Installation' is shown, followed by '1.1. Raspbian installation'. The text explains that if using the Raspbian distro, picamera is likely installed by default and can be tested by running 'python -c "import picamera"' or 'python3 -c "import picamera"'. A code block shows the successful execution of these commands. Below this, it states that if no error occurs, picamera is installed, and if an error does, it shows a sample traceback. Another code block shows the error message: 'ImportError: No module named 'picamera''. Finally, it provides instructions on how to install picamera on Raspbian using the system's package manager 'apt'.

Picamera
release-1.13

Search docs

- 1. Installation
- 2. Getting Started
- 3. Basic Recipes
- 4. Advanced Recipes
- 5. Frequently Asked Questions (FAQ)
- 6. Camera Hardware
- 7. Development
- 8. Deprecated Functionality
- 9. API - The PiCamera Class
- 10. API - Streams
- 11. API - Renderers
- 12. API - Encoders
- 13. API - Exceptions
- 14. API - Colors and Color Matching
- 15. API - Arrays
- 16. API - mmalobj
- 17. Change log
- 18. License

Read the Docs v: release-1.13

Docs » 1. Installation Edit on GitHub

1. Installation

1.1. Raspbian installation

If you are using the Raspbian distro, you probably have picamera installed by default. You can find out simply by starting Python and trying to import picamera:

```
$ python -c "import picamera"
$ python3 -c "import picamera"
```

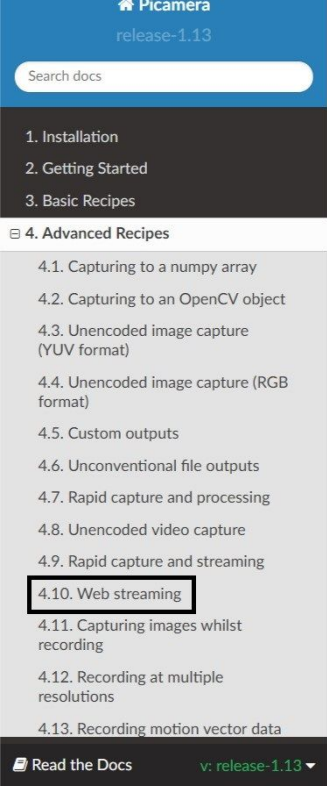
If you get no error, you've already got picamera installed! Just continue to Getting Started. If you don't have picamera installed you'll see something like the following:

```
$ python -c "import picamera"
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ImportError: No module named picamera
$ python3 -c "import picamera"
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ImportError: No module named 'picamera'
```

To install picamera on Raspbian, it is best to use the system's package manager: apt. This will ensure that picamera is easy to keep up to date, and easy to remove should you wish to do so. It will also make picamera available for all users on the system. To install picamera using apt simply run:

3ο Βήμα

Πηγαίνουμε στην υποπαράγραφο 4.10 (Web Streaming)



The screenshot shows the Picamera documentation website. The sidebar on the left lists the table of contents, with '4.10. Web streaming' highlighted with a black box. The main content area shows the '4. Advanced Recipes' section, which includes a warning about naming files and the '4.1. Capturing to a numpy array' section.

4. Advanced Recipes

The following recipes involve advanced techniques and may not be "beginner friendly". Please feel free to suggest enhancements or additional recipes.

Warning

When trying out these scripts do *not* name your file `picamera.py`. Naming scripts after existing Python modules will cause errors when you try and import those modules (because Python checks the current directory before checking other paths).

4.1. Capturing to a numpy array

Since 1.11, picamera can capture directly to any object which supports Python's buffer protocol (including numpy's `ndarray`). Simply pass the object as the destination of the capture and the image data will be written directly to the object. The target object must fulfil various requirements (some of which are dependent on the version of Python you are using):

1. The buffer object must be writeable (e.g. you cannot capture to a `bytes` object as it is immutable).
2. The buffer object must be large enough to receive all the image data.
3. (Python 2.x only) The buffer object must be 1-dimensional.
4. (Python 2.x only) The buffer object must have byte-sized items.

For example, to capture directly to a three-dimensional numpy `ndarray` (Python 3.x only):

4ο Βήμα

Αντιγράφουμε τον κώδικα που μας δίνει σε έναν επεξεργαστή κειμένου (πχ. notepad++)

BabySpy - Ενδοεπικοινωνία Μωρού

```
import io
import picamera
import logging
import socketserver
from threading import Condition
from http import server

PAGE="""\
<html>
<head>
<title>BabySpy Camera</title>
</head>
<body>
<h1>BabySpy Camera</h1>

</body>
</html>
"""

class StreamingOutput(object):
    def __init__(self):
        self.frame = None
        self.buffer = io.BytesIO()
        self.condition = Condition()

    def write(self, buf):
        if buf.startswith(b'\xff\xd8'):
            # New frame, copy the existing buffer's content and notify all
            # clients it's available
            self.buffer.truncate()
            with self.condition:
                self.frame = self.buffer.getvalue()
                self.condition.notify_all()
            self.buffer.seek(0)
        return self.buffer.write(buf)

class StreamingHandler(server.BaseHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/':
            self.send_response(301)
            self.send_header('Location', '/index.html')
            self.end_headers()
        elif self.path == '/index.html':
            content = PAGE.encode('utf-8')
            self.send_response(200)
```

BabySpy - Ενδοεπικοινωνία Μωρού

```
        self.send_header('Content-Type', 'text/html')
        self.send_header('Content-Length', len(content))
        self.end_headers()
        self.wfile.write(content)
    elif self.path == '/stream.mjpg':
        self.send_response(200)
        self.send_header('Age', 0)
        self.send_header('Cache-Control', 'no-cache, private')
        self.send_header('Pragma', 'no-cache')
        self.send_header('Content-Type', 'multipart/x-mixed-replace;
boundary=FRAME')
        self.end_headers()
        try:
            while True:
                with output.condition:
                    output.condition.wait()
                    frame = output.frame
                self.wfile.write(b'--FRAME\r\n')
                self.send_header('Content-Type', 'image/jpeg')
                self.send_header('Content-Length', len(frame))
                self.end_headers()
                self.wfile.write(frame)
                self.wfile.write(b'\r\n')
            except Exception as e:
                logging.warning(
                    'Removed streaming client %s: %s',
                    self.client_address, str(e))
        else:
            self.send_error(404)
            self.end_headers()

class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):
    allow_reuse_address = True
    daemon_threads = True

with picamera.PiCamera(resolution='640x480', framerate=24) as camera:
    output = StreamingOutput()
    camera.start_recording(output, format='mjpeg')
    try:
        address = ('', 8000)
        server = StreamingServer(address, StreamingHandler)
        server.serve_forever()
    finally:
        camera.stop_recording()
```


4ο Βήμα

Αποθηκεύουμε το αρχείο με ένα όνομα της αρεσκείας μας, με κατάληξη .py και το εκτελούμε

5ο Βήμα

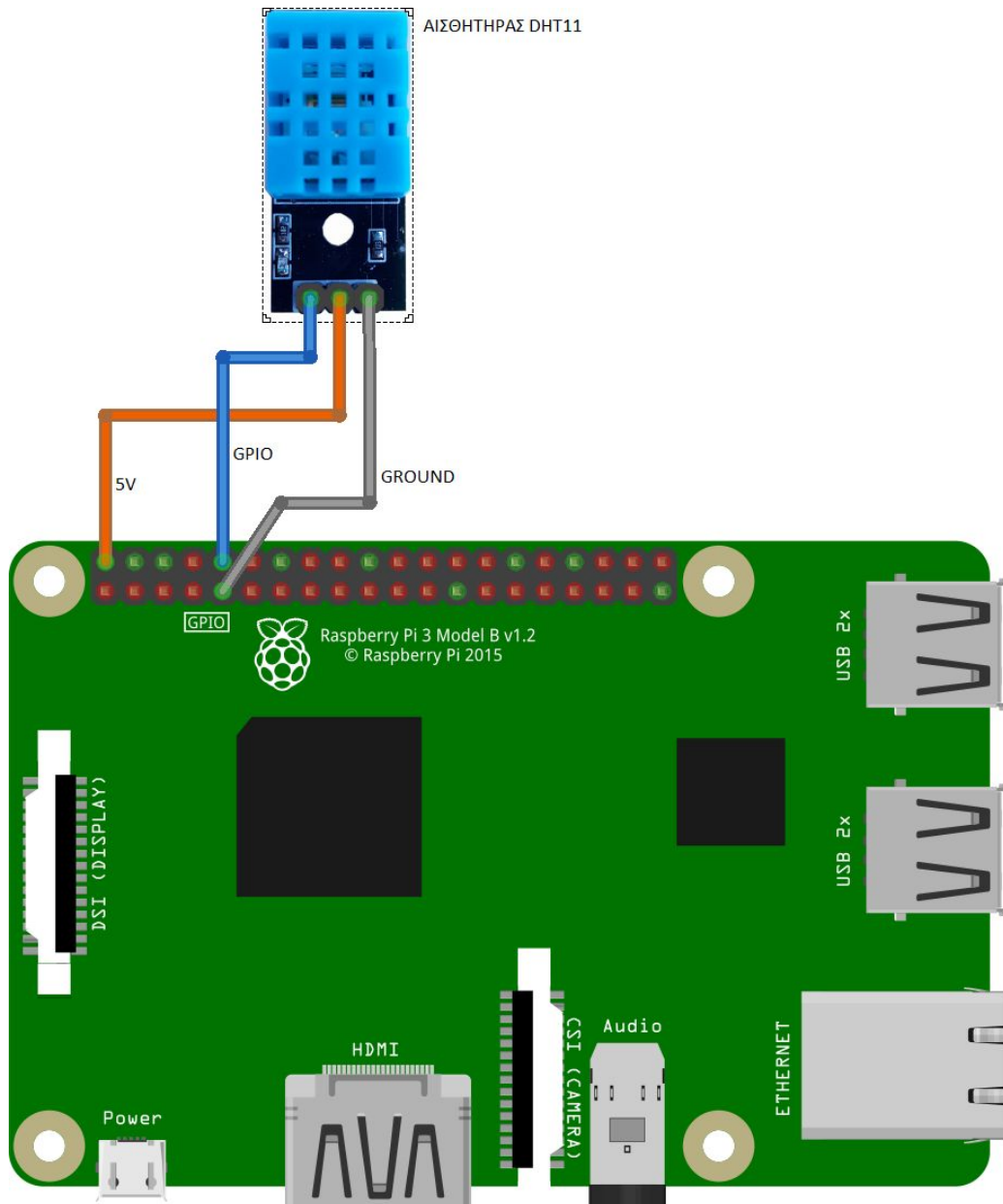
Αφού εκτελέσουμε τον κώδικα μας , ανοίγουμε έναν browser και πληκτρολογούμε την διεύθυνση :

<http://pi-address:8000/>

Όπου pi-address η διεύθυνση τοπικού δικτύου του Raspberry Pi.

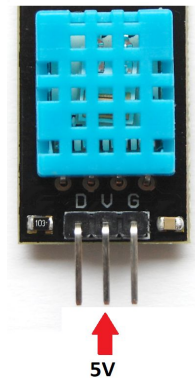
Προετοιμασία του αισθητήρα θερμοκρασίας και υγρασίας (DHT11)

Συνδεσμολογία Αισθητήρα



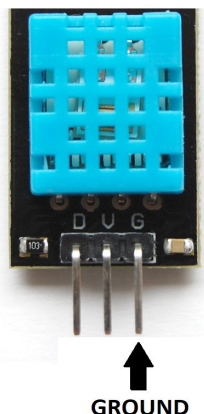
1) Σύνδεση καλωδίου Ρεύματος (5V)

Απο την μια άκρη του θηλυκού καλωδίου (πορτοκαλί χρώμα) συνδέουμε το Raspberry σε ένα pin ρεύματος (5V) και την άλλη άκρη την συνδέουμε στο μεσαίο pin του DHT11.



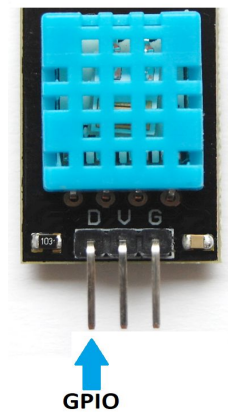
2) Σύνδεση καλωδίου Γείωσης (GROUND)

Απο την μια άκρη του θηλυκού καλωδίου (γκρι χρώμα) συνδέουμε το Raspberry σε ένα pin γείωσης (GND) και την άλλη άκρη την συνδέουμε στο δεξί pin του DHT11.



3) Σύνδεση καλωδίου δεδομένων (GPIO)

Απο την μια άκρη του θηλυκού καλωδίου (μπλέ χρώμα) συνδέουμε το Raspberry στο pin δεδομένων (GPIO) και την άλλη άκρη την συνδέουμε στο αριστερό pin του DHT11.



Μόλις τελειώσουμε την σύνδεση των καλωδίων πάμε στο προγραμματιστικό κομμάτι.

Προγραμματισμός του Raspberry

Πριν Ξεκινήσουμε να γράφουμε τον κώδικα μας , είναι απαραίτητο να εγκαταστήσουμε τις απαραίτητες Βιβλιοθήκες (Libraries).

Για το προγραμματισμό του αισθητήρα DHT11 θα χρειαστούμε τις παρακάτω βιβλιοθήκες :

- Adafruit_DHT | Προεγκατεστημένη
- time | Προεγκατεστημένη

Όλες οι απαραίτητες βιβλιοθήκες είναι ήδη εγκατεστημένες

Προγραμματισμός του αισθητήρα

Προγραμματισμός Raspberry για εμφάνιση στην οθόνη την θερμοκρασία και την υγρασία του δωματίου

Καλό είναι να δημιουργήσουμε έναν φάκελο όπου θα βάλουμε εκεί όλα τα αρχεία κώδικα.

1ο Βήμα

Σε έναν επεξεργαστή κειμένου (πχ. notepad++)
αντιγράφουμε τον παρακάτω κώδικα:

```
#εισαγωγή απαιτούμενων βιβλιοθηκών
import Adafruit_DHT
import time

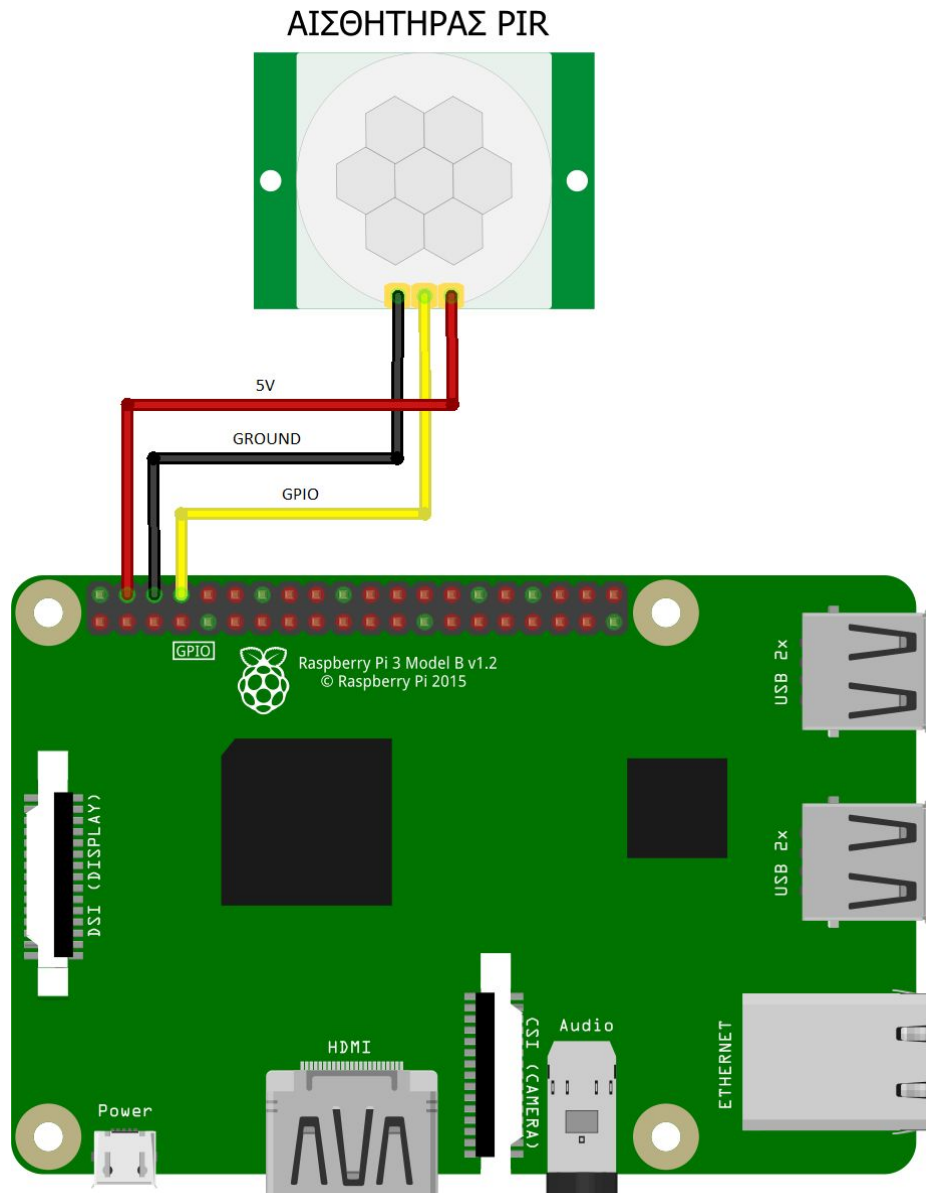
while True:
    #δηλώνουμε τον αισθητήρα
    sensor = Adafruit_DHT.DHT11
    #δηλώνουμε την θέση στην οποία συνδέσαμε τον αισθητήρα θερμοκρασίας
    pin = 15
    #δημιουργούμε μεταβλητες για την υγρασία και τη θερμοκρασία
    humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
    #Λέμε στο πρόγραμμα να κάνει παύση για 4 δευτερόλεπτα
    time.sleep(5)
    #έλεγχος για αποτυχία
    if humidity is not None and temperature is not None:
        #τύπωσε στην οθόνη την θερμοκρασία και την υγρασία
        print('Θερμοκρασία={0:0.1f}*C Υγρασία={1:0.1f}%'.format(temperature, humidity))
    else:
        print('Αποτυχία!')
```

2ο Βήμα

Αποθηκεύουμε το αρχείο με ένα όνομα της αρεσκείας μας,
με κατάληξη .py και το εκτελούμε.

Προετοιμασία του αισθητήρα κίνησης (PIR Sensor)

Συνδεσμολογία Αισθητήρα



1) Σύνδεση καλωδίου Ρεύματος (5V)

Απο την μια άκρη του θηλυκού καλωδίου (κόκκινο χρώμα) συνδέουμε το Raspberry σε ένα pin ρεύματος (5v) και την άλλη άκρη την συνδέουμε στο δεξί pin του PIR.



2) Σύνδεση καλωδίου Γείωσης (GROUND)

Απο την μια άκρη του θηλυκού καλωδίου (μαύρο χρώμα) συνδέουμε το Raspberry σε ένα pin γείωσης (GND) και την άλλη άκρη την συνδέουμε στο αριστερό pin του PIR.



3) Σύνδεση καλωδίου δεδομένων (GPIO)

Απο την μια άκρη του θηλυκού καλωδίου (κίτρινο καλώδιο) συνδέουμε το Raspberry στο pin δεδομένων (GPIO) και την άλλη άκρη την συνδέουμε στο μεσαίο pin του PIR.



Μόλις τελειώσουμε την σύνδεση των καλωδίων πάμε στο προγραμματιστικό κομμάτι.

Προγραμματισμός του Raspberry

Πριν Ξεκινήσουμε να γράφουμε τον κώδικα μας , είναι απαραίτητο να εγκαταστήσουμε τις απαραίτητες Βιβλιοθήκες (Libraries).

Για το προγραμματισμό του αισθητήρα PIR θα χρειαστούμε τις παρακάτω βιβλιοθήκες :

- gpiozero | Website : <https://gpiozero.readthedocs.io/en/stable/>
- signal | Προεγκατεστημένη
- time | Προεγκατεστημένη

Εγκατάσταση της βιβλιοθήκης gpiozero

- 1) Ανοίγουμε το τερματικό
- 2) Ενημερώνουμε τη λίστα αποθετηρίων σας:

```
sudo apt update
```

- 3) Στη συνέχεια, εγκαθιστούμε το πακέτο για το Python 3:

```
sudo apt install python3-gpiozero
```

Προγραμματισμός του αισθητήρα

Προγραμματισμός του αισθητήρα για εμφάνιση στην οθόνη εάν βρέθηκε κίνηση στο δωμάτιο

Καλό είναι να δημιουργήσουμε έναν φάκελο όπου θα βάλουμε εκεί όλα τα αρχεία κώδικα.

1ο Βήμα

Σε έναν επεξεργαστή κειμένου (πχ. notepad++)
αντιγράφουμε τον παρακάτω κώδικα:

```
#εισαγωγή απαιτούμενων βιβλιοθηκών
from gpiozero import MotionSensor
from signal import pause
import time

#δηλώνουμε την θέση στην οποία συνδέσαμε το PIR
pir=MotionSensor(14)
#για να δημιουργήσουμε μια συνεχόμενη επανάληψη
#χρησιμοποιούμε το while True
while True:
    #Λέμε στο πρόγραμμα να κάνει παύση για 4 δευτερόλεπτα
    time.sleep(4)
    #έλεγχος εάν βρέθηκε κίνηση
    if pir.motion_detected:
        #κίνηση βρέθηκε
        print ('Βρέθηκε Κίνηση')
    else:
        #κίνηση δεν βρέθηκε
        print ('Δεν Βρέθηκε Κίνηση')
```

2ο Βήμα

Αποθηκεύουμε το αρχείο με ένα όνομα της αρεσκείας μας,
με κατάληξη .py και το εκτελούμε