

Міністерство освіти і науки України  
Чорноморський національний університет імені Петра Могили  
Факультет комп'ютерних наук  
Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО  
ЗАХИСТУ  
Завідувач кафедри,  
канд. тех. наук, доцент

Я.М. Крайник
« _ » _____ 2022 р.

МАГІСТЕРСЬКА РОБОТА

Галузь знань: 12 Інформаційні технології  
Спеціальність: 123 Комп'ютерна інженерія  
Тема: Система детектування руху на основі ESP8266-  
SAM та Raspberry Pi  
Шифр: 123 – МР.ПЗ.00 – 605.1611507

Виконав:

студент 6 курсу, групи 605,  
спеціальності  
123 Комп'ютерна  
інженерія  
Р. В. Ілляшенко

Керівник:

канд. фіз-мат. наук, доцент  
С. В. Пузирьов

Миколаїв 2022

Кафедра комп'ютерної інженерії  
Система детектування руху на основі ESP8266-CAM та Raspberry Pi

## АНОТАЦІЯ

Магістерської роботи

«Система детектування руху на основі ESP8266-CAM та Raspberry Pi»

Студент: Ілляшенко Руслан Віталійович

Керівник: Доцент Пузирьов С. В.

Магістерська робота спрямована на створенні системи детектування руху на основі Raspberry Pi. Практичне значення результатів дослідження та розроблення полягає у створенні системи детектування руху. В якості мови реалізації програми обрано мову Python.

Пояснювальна записка магістерської роботи складається зі вступу, трьох розділів, висновків та одного додатка.

У вступі визначається актуальність теми, сформульовані мета, об'єкт, предмет та завдання дослідження та розроблення магістерської роботи.

У першому розділі розглянуті поставлені задачі магістерської роботи; вибір одноплатного ком'ютера.

У другому розділі проводиться аналіз розгляд апаратної частини пристрою для детектування та запису; опис використаних елементів; наведені блок-схеми алгоритмів.

У третьому розділі розглянуто програмну частину системи детектування.

У висновках наведено аналіз виконаної роботи та отриманих результатів дослідження та розроблення.

У додатку А наведений вихідний код програми.

В цілому, магістерська робота без додатків містить 62 сторінок, 50 рисунків, 1 таблиць, 15 джерел посилання, 1 блок-схеми.

## ABSTRACT

Of the Bachelor's Thesis

«ESP8266-CAM and Raspberry Pi motion detection system»

Student: Illiashenko Ruslan Vitaliyovych

Consultant: Associate Professor of CE Puzyrov S.V.

The master's thesis is aimed at creating a motion detection system based on the Raspberry Pi. The practical significance of the results of research and development is to create a motion detection system. The Python language is selected as the program implementation language.

The explanatory note of the master's thesis consists of an introduction, three chapters, conclusions and one appendix.

The introduction defines the topicality of the topic, formulated purpose, object, subject and tasks of research and development of master's thesis.

In the first section the tasks of the master's work are considered; choosing a single board computer.

The second section analyzes the analysis of the hardware of the device for detection and recording; description of used elements; block diagrams of algorithms are given.

The third section discusses the software part of the detection system.

The conclusions provide an analysis of the work performed and the results of research and development.

Appendix A provides the source code of the program.

In total, the master's thesis without appendices contains 62 pages, 50 figures, 1 table, 15 reference sources, 1 flowchart.

## ЗМІСТ

ВСТУП .....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1 Детектування руху для розумного будинку.....	10
1.2 Особливості мікрокомп'ютера Raspberry Pi.....	11
1.3 Способи і технології реалізації системи детектування руху.....	13
1.4 Вибір камери для запису.....	14
Висновок до першого розділу.....	18
2 АПАРАТНА ЧАСТИНА.....	19
2.1 Пристрій системи детектування руху.....	19
2.2 Опис використаних елементів.....	19
2.2.1 Raspberry Pi A+.....	20
2.2.2 Raspberry Pi Camera Board.....	22
2.2.3 Датчик руху PIR.....	24
2.3 Схема підключення.....	25
2.4 Принцип роботи.....	27
2.5 Модуль ESP8266.....	28
2.5.1 Характеристики:.....	29
2.5.2 Контакти:.....	29
2.5.3 Способи роботи.....	30
2.5.4 Програмування ESP8266.....	30
2.6 Модуль ESP32-CAM та потокова передача відео.....	30
2.6.1 Основні характеристики ESP32-CAM.....	31
2.6.2 Алгоритм потокової передачі відео з використанням WebSocket.....	32
2.6.3 Керуюча програма для отримання потокового відео з ESP32-CAM..	33

Висновок до другого розділу .....	39
3 ПРОГРАМНА ЧАСТИНА.....	40
3.1 Опис алгоритму роботи: .....	40
3.2 Raspberry Pi перший запуск.....	40
3.3 Запис образу Raspbian на карту пам'яті.....	41
3.4 Підключення Raspberry Pi.....	41
3.5 Налаштування Raspberry Pi.....	41
3.6 Корисні консольні програми для Raspberry Pi.....	48
3.7 Команди Linux .....	50
3.8 Резервне копіювання Raspberry Pi.....	51
3.9 Тестування та налаштування модуля камери. ....	51
3.10 Створення та підключення схеми. ....	52
3.11 Розширені налаштування камери .....	54
3.12 Область застосування міні-ПК Raspberry Pi .....	59
Висновок до третього розділу.....	61
ВИСНОВКИ.....	62
Додаток А. Вихідний код програми .....	63
ПЕРЕЛІК ПОСИЛАНЬ .....	64

## ВСТУП

Raspberry Pi побудовано на системі Broadcom BCM2835 chip-on-chip (SoC), яка включає процесор ARM 700 МГц, графічний процесор VideoCore IV і 512 або 256 мегабайт оперативної пам'яті. Жорсткого диска немає, замість нього використовується SD-карта. Це обладнання дозволяє відтворювати відео H.264 з роздільною здатністю 1080p і запускати комп'ютерні ігри, як-от Quake III Arena.

Ініціатором проекту Raspberry Pi є британська благодійна організація Raspberry Pi Foundation. Комп'ютер був розроблений як пристрій для навчання дітей програмуванню, але він здобув популярність і в інших сферах – зокрема, він використовується для створення домашніх медіацентрів. Найдешевший Raspberry Pi поставляється без футляра і виглядає як плата розміром з кредитну картку. Плата важить 45 грам. Комп'ютер використовує процесор 700 МГц на основі архітектури ARM; є роз'єм для навушників і слот для карти пам'яті. Молодша (A) і старша (B) моделі Raspberry Pi відрізняються за обсягом оперативної пам'яті (256 і 512 мегабайт відповідно) і кількістю портів USB.

Старий Raspberry Pi надійшов у продаж наприкінці лютого 2012 року за 35 доларів. Молодша модель Raspberry Pi за 25 доларів надійшла в продаж у Європі в лютому 2013 року. Комп'ютери Raspberry Pi можна придбати у дистриб'юторів RS Components та element14. З лютого 2012 року було продано понад мільйон пристроїв.

Raspberry Pi має спеціалізований дистрибутив Linux, ОС Raspbian (на основі дистрибутива Debian) і Pi Store, який пропонує платні та безкоштовні програми.

На початку 2020 року Canonical випустила Ubuntu Server 20.04 Focal Fossa з офіційною підтримкою Raspberry Pi 4. Пізніше, у жовтні того ж року, підтримка Raspberry Pi 4 була додана для спеціальної версії Ubuntu версії 20.10 Groovy Gorilla.

Актуальність - Впроваджуючи інформаційно-технічні засоби в повсякденне життя, людство створює для себе комфорт і економію

Об'єкт дослідження - система детектування руху.

Предмет дослідження - Програмно-апаратний комплекс для детектування руху, налаштування мікрокомп'ютера для автоматизації системи.



Кафедра комп'ютерної інженерії  
Система детектування руху на основі ESP8266-CAM та Raspberry Pi

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Для досягнення поставленої мети необхідне вирішити наступні завдання:

- розглянути можливість придбання необхідного обладнання;
- навчитися користуватися Raspberry Pi, працювати з її бібліотеками, розуміти і коригувати вивчити літературу по створенню системи детектування руху;
- програми для своїх завдань;
- створити прототип системи детектування і провести серію первинних випробувань з налагодження її роботи;
- дослідити оптимальні параметри функціонування системи на рух;
- дати рекомендації щодо можливості системи детектування в умовах тривалої відсутності руху в квартирі.

У своїй роботі використані наступні методи дослідження:

- аналіз і синтез інформації з Internet-ресурсів, спеціальної літератури, що розкриває суть даного завдання;
- моделювання системи детектування;
- проведення дослідницьких робіт з виявлення оптимальних умов функціонування системи.

### 1.1 Детектування руху для розумного будинку

Зараз масово впроваджують розумну автоматизацію в будинок.

На рисунку 1.1 зображено розумний будинок



Рисунок 1.1 Розумний будинок

Розумний будинок – це дуже складна система датчиків, механічних та електронних компонентів, керована за закладеною програмою. Ця система стежить за витратою води, електрики, управляє освітленням, захищає від несанкціонованого доступу, забезпечує віддалене управління різними пристроями по мобільному телефону або Інтернету. Основне завдання таких систем – полегшити життя, доручивши частину домашньої роботи мікроконтролера.

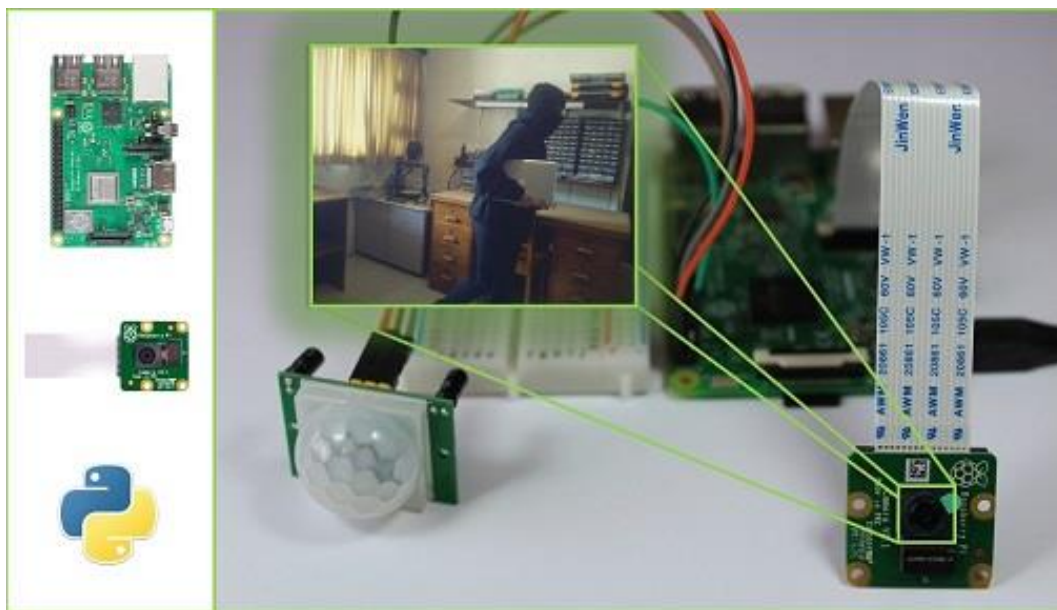


Рисунок 1.2 Детектор руху

Детектор руху – частина цієї системи. На рисунку 1.1 зображений детектор руху. Детектор може бути влаштований дуже просто

## 1.2 Особливості мікрокомп'ютера Raspberry Pi

Одноплатний комп'ютер Raspberry Pi — це невеликий гаджет у формі відкритої плати, розмір якого відповідає банківській картці. Має процесор з оперативною пам'яттю, роз'єми для HDMI і MIPI DSI, 2 порти для периферійних пристроїв USB версії 2.0, роз'єм RCA для відеообладнання, роз'єм 3,5 мм для підключення до аудіотехніки, порт Ethernet і слот для підключення карт пам'яті SD.

Переваги мікрокомп'ютера Raspberry:

- Перш за все – ціна, яка варіюється від 5 до 35 доларів в залежності від моделі. Найкращі варіанти – це Zero і Zero W (5 і 10 доларів відповідно), які відрізняються від інших раміанів Raspberry Pi від інших 1

- Процесор і відеоприскорювач Raspberry Pi відповідають характеристикам портативних пристроїв за сотні доларів і забезпечують високу продуктивність у рідних ОС і системах на базі Linux;
- Підтримка низки операційних систем: Raspbian - вбудована операція розробника, Pidora - операція розробки на базі Linux, Kodi - обмежене відтворення медіа, OSMC - медіаплеєр з відкритим кодом, RISC OS - "операція" процесора розробки, Windows 10 - в описі не обов'язково, тільки для моделей 2B і 3B. Окрім офіційної підтримки на Raspberry у 2017 році, він зумів успішно запуснути більше десятка інших невідновлюваних джерел енергії;
- Детальна конфігурація системи при першому запуску з Raspberry Config. Він дозволяє розганяти частоту процесора, включити підтримку камери, змінити положення. Надалі ви можете перезапустити його в будь-який момент;
- Країна-виробник - Великобританія. Тому, замовляючи на Aliexpress Raspberry, ви можете не переживати за якість пристрою, адже створення подібних цілий рік. Крім того, якщо ваш домашній комп'ютер перестав працювати, і ви знаєте причину проблеми, просто купіть бортовий комп'ютер raspberry pi і спробуйте замінити пошкоджену частину.

Виходячи з цих переваг, не дивно, що високі продажі міні-комп'ютерів Raspberry Pi. Залишилося пояснити, де краще використовувати гаджет.

Магістерська робота виконана на версії Raspberry Pi 3. На рисунку 1.3 зображено мікрокомп'ютер Raspberry Pi 3.

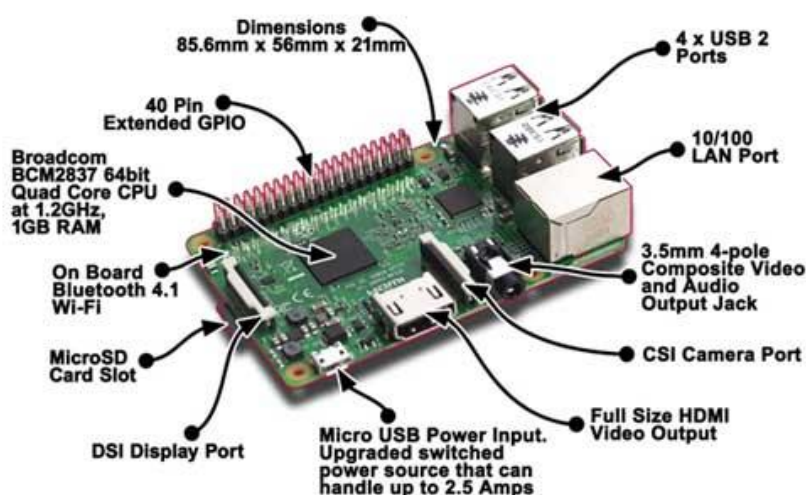


Рисунок 1.3 – Raspberry Pi 3

При першому включенні необхідно:

1. Raspberry Pi
2. USB клавіатура
3. USB мишка
4. Від 2 до 32 ГБ SD / MMC / SDIO карта пам'яті (з 6 класів до 10 класу)
5. Micro USB-адаптер, щоб увімкнути Raspberry Pi
6. Монітор з HDMI або телевізором або ТВ-тюнером із RCA-контактом
7. Кабель HDMI для підключення до монітору або RCA кабель для підключення до ТВ або ТВ-тюнер
8. Win32DiskImager програма для того, щоб записати образ на флеш карту.
9. Образ операційної системи Raspberry Pi "wheezy".
10. Картрідер для считування SD/MMC/SDIO карт пам'яті.

### **1.3 Способи і технології реалізації системи детектування руху.**

У розділі розглянуті способи і технології реалізації системи детектування руху.

Функціональні характеристики ідеальної системи детектору залежать від багатьох зовнішніх факторів, до яких можна віднести наступні:

- дальність обробки датчика руху;
- якість на фокус камери;
- освітлення;
- особливості середовища де буде використовуватись датчик руху.

Таким чином, для оптимальної роботи запису відео під час роботи датчика необхідно врахувати два параметри, залежних від вище перерахованих факторів:

- розміри приміщення де буду використовуватись мікрокомп'ютер;
- та час та відстань за якими буду працювати датчик руху.

Під час аналізу літератури було визначено, що Raspberry Pi 3 дозволяє використати широкий спектр відносно дешевих датчиків та приладів, а також програмувати складні алгоритми детектування, які потребуються для детектування у різних середовищах.

### 1.4 Вибір камери для запису.

В ході виконання магістерської роботи був проведений огляд існуючих систем детектування руху. Основні системи наведені в таблиці 1. Крім того, в таблиці 1 наведені ціни та магазини в яких можна придбати систему детектування.

Назва систем	Ці на систем	Характеристики	Магазин
Raspberry Pi Camera Board Sony IMX219 Рисунок - 1.4	\$ 44.07	Високоякісна 8 мегапіксельний датчик зображення Sony IMX219 з фіксуючим фокусом. Датчик підтримує зображення з розширенням 3280 x 2464 пікселя, а також відео 1080p30, 720p60 та 640x480p60/90. Підключається до плати завдяки невеликому роз'єму на верхній частині плати та використовує особливий CSI інтерфейс. Сам датчик невеликий, всього 25мм x 23мм x 9мм.	ev o.net
Waveshare IMX477-160 12.3MP Camera Рисунок - 1.5	\$ 9.97	Дозвіл 12МПикс. (4056x3040). Матриця: Sony IMX477 Розмір сонсору: 7,9мм. Розмір пікселя: 1.55µm (H) × 1.55µm (V) Кут зору: 160°(D) 118°(H) 87°(V) Робоча напруга: 3,3 В	ev o.net
Waveshare IMX219-160 8MP IR-CUT Camera Рисунок - 1.6	\$ 34.95	Модуль камери на базі сенсору IMX219, з роздільною здатністю 8МП, з інфрачервоним світлодіодами підсвічування IR-CUT, з кутом огляду 162°. Цей модуль забезпечує розпізнавання облич, дорожніх знаків, розпізнавання номерних знаків і т.д. Розширення: 3280 x 2464. Кут зору (FOV): 162,4 ° (D) 124 ° (H) 91 ° (V) Робоча напруга: 3,3 В Розміри: 30.4x79.7x19.95	ev o.net
Waveshare IMX219-83 Stereo Camera Рисунок - 1.7	\$ 59.50	Стереокамера, яка має 2 камери IMX219 на борту з роздільною здатністю 8МП Дозвіл: 3280 x 2464 (на кожному камеру) Фокусна відстань: 2,6 мм Кут огляду: 83/73/50 градусів (діагональ/горизонталь/вертикаль)	ev o.net

Кафедра комп'ютерної інженерії  
Система детектування руху на основі ESP8266-CAM та Raspberry Pi

		Розміри: 24/85/60(між камерами)	
Waveshare RPi Camera Рисунок - 1.8	\$ 27.42	Модуль камери з функцією ночного бачення. Модуль має встроєний світлодіод, працюючий в інфракрасному діапазоні для «підсвітки» оточення. 5 Мп сенсор OV5647. Розміри: 31x32мм	ev o.net

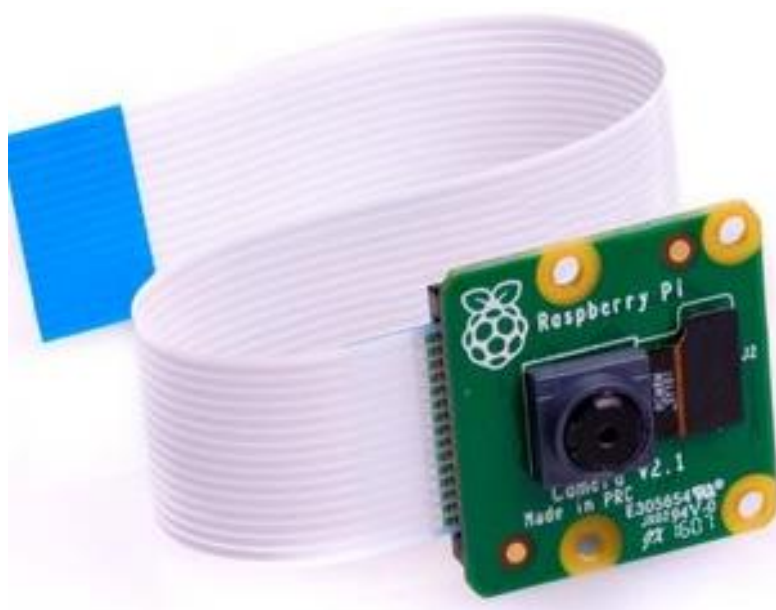


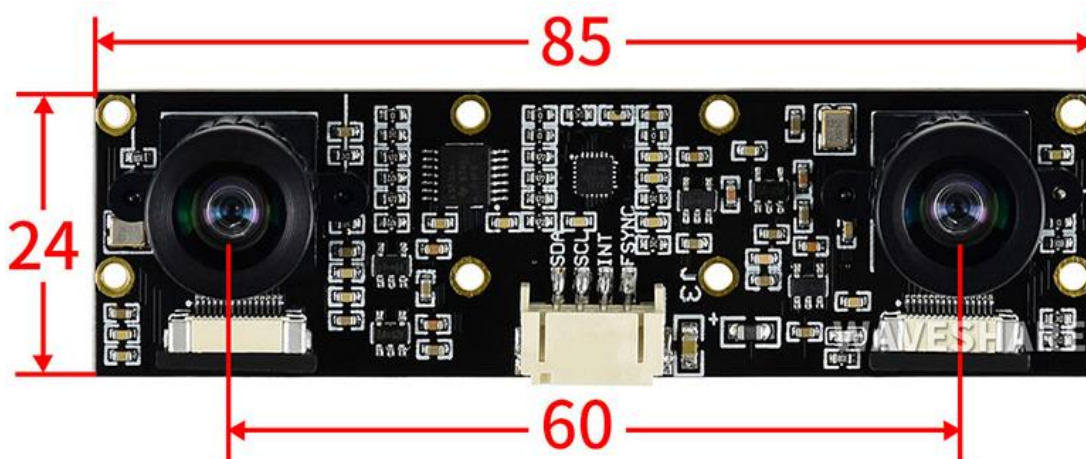
Рисунок 1.4 - Raspberry Pi Camera Board Sony IMX219



Рисунок 1.5 - Waveshare IMX477-160 12.3MP Camera



Рисунок 1.6 - Waveshare IMX219-160 8MP IR-CUT Camera



**Unit:mm**

Рисунок 1.7 - Waveshare IMX219-83 Stereo Camera



Рисунок 1.8 - Waveshare RPi IR-CUT Camera

Таким чином, ми можемо обрати камеру для наших потреб:

- забезпечити кращу та якісну запис;



- керувати системою через зручний інтерфейс, доступний на персональному комп'ютері або смартфоні;
- проводити збір, аналіз та обробку даних з камери в будь який час доби.

Беручи до уваги недоліки існуючих систем, актуальною є задача розробка системи детектування руху з урахуванням поточного освітлення навколишнього середовища.

### **Висновок до першого розділу**

У цьому розділі магістерської роботи було розглянено методи полегшення життя. Особливу увагу приділено вибору камери для проекту. Розглянуто особливості та види Raspberry Pi. Розглянуто що потрібно для першого включення мікрокомп'ютера. А також розглянуто для чого потрібна система детектування.

## 2 АПАРАТНА ЧАСТИНА

### 2.1 Пристрій системи детектування руху.

На рис 2.1 зображено блок–схему візуальних компонентів, які створюють функціональний і водночас простий та зрозумілий інтерфейс. Даний інтерфейс є простим і водночас легким для сприйняття користувача.

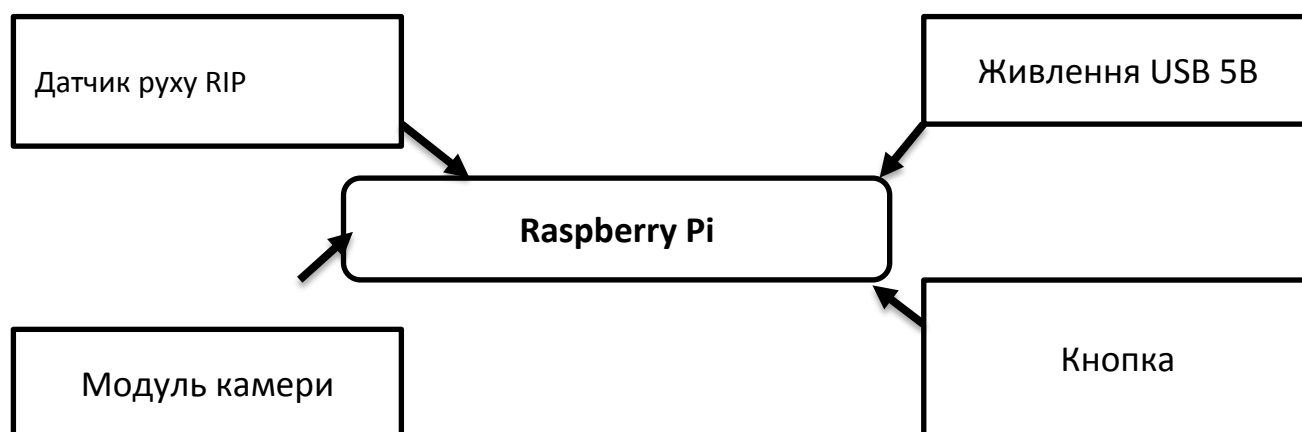


Рисунок 2.1 – Пристрій системи детектування руху

Представлена наступна блок–схема системи. Для її реалізації було придбано необхідні компоненти: мікрокомп'ютер, модуль камери, живлення 5В, датчики руху та ін. Після запуску пристрою у безперервному робочому циклі опитується датчик руху і виходячи зі стану датчика, виконуються дії. Датчик руху керує камерою.

### 2.2 Опис використаних елементів

Система детектування руху - незамінний помічник, як для огляду свого будинку в середині, так і зовні. Система включає камеру для запису зображення, якщо повз датчик руху пройшов об'єкт. Камера може вимикатись та вмикатись завдяки кнопки.

**Для виготовлення робочої системи детектування необхідні наступні компоненти:**

1. Raspberry Pi - 1шт.
2. PIR датчика руху - 1шт.
3. Кнопка - 1шт.
4. Модуль камера - 1шт.

## 5. Блок живлення 5В - 1шт.

Далі доречно буде зробити більш детальний опис основних використаних елементів для побудови системи детектування руху на основі мікрокомп'ютера Raspberry Pi.

### 2.2.1 Raspberry Pi A+

Raspberry Pi 3 A+ Якщо Ви вперше проектуєте на Raspberry Pi, покупка такої плати буде оптимальним варіантом.

На рис 2.2 представлено зображення Raspberry Pi 3 A+



Рисунок 2.2 - Raspberry Pi 3 A+

Загальні відомості:

Raspberry Pi 3 Model A+ є останньою розробкою Raspberry Pi Foundation і є мініатюрною версією повноцінної Raspberry Pi 3 Model B+ на базі того ж чотириядерного 64-бітного процесора Broadcom BCM2837B0, 512 МБ ОЗУ, дводіапазонного Wi-Fi та Bluetooth 4.2 BLE. Новий Raspberry Pi 3 A+ не має порту Ethernet і замість звичайних 4 портів USB має лише один. Однак він має менші габарити (65x56 мм) та менше енергоспоживання. У плату також було внесено деякі зміни, спрямовані на покращення завантаження USB та управління температурним режимом. Карта повністю сумісна з усіма пристроями та програмним забезпеченням.

У таблиці 2.1 наведені характеристики Arduino Uno

Кафедра комп'ютерної інженерії  
Система детектування руху на основі ESP8266-CAM та Raspberry Pi

Таблиця 2.1 – Технічні характеристики Raspberry Pi 3 Model A+

<b>RAM</b>	<b>512MB LPDDR2</b>
<b>Графічний процесор</b>	<b>VideoCore IV</b>
<b>Wi-Fi</b>	<b>(2.4GHz / 5GHz)</b>
<b>Bluetooth</b>	<b>4.2 LE (чип Cypress CYW43455)</b>
<b>Слот для карт microSD</b>	<b>1</b>
<b>USB 2.0</b>	<b>1</b>
<b>HDMI</b>	<b>1</b>
<b>Вивід звуку</b>	<b>3.5 jack</b>
<b>GPIO</b>	<b>40 пінов</b>
<b>Інтерфейс для підключення камери</b>	<b>CSI</b>
<b>Інтерфейс для підключення дисплея</b>	<b>DSI</b>
<b>Живлення</b>	<b>5V 2.4 A через microUSB роз'єм.</b>

### 2.2.2 Raspberry Pi Camera Board

Модуль камери Raspberry Pi v2 — це високоякісний 8-мегапіксельний датчик зображення Sony IMX219 з фіксованим фокусом, розроблений спеціально для плати Raspberry Pi.

На рис 2.3 представлено зображення Raspberry Pi Camera Board.

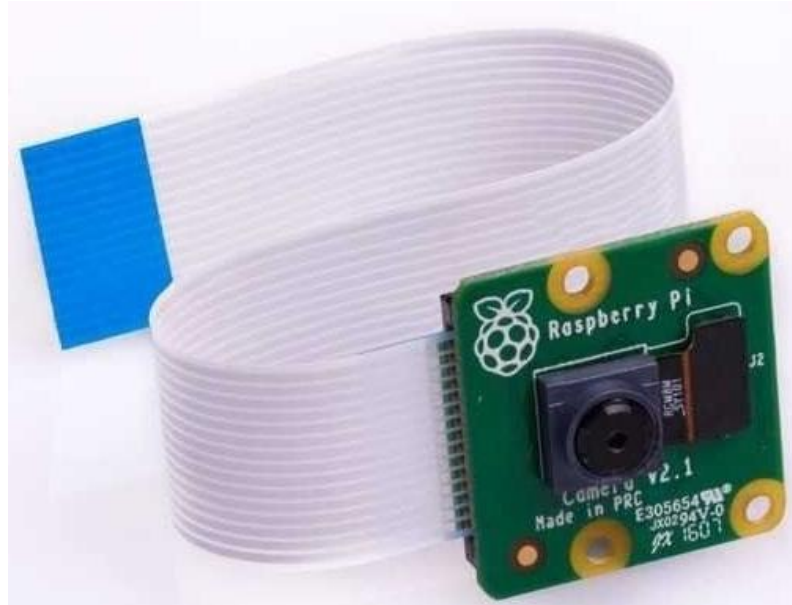


Рисунок 2.3 - Raspberry Pi Camera Board

Загальні відомості:

Датчик підтримує зображення з роздільною здатністю 3280 x 2464 пікселів, а також 1080p30, 720p60 і відео 640x480p60 / 90. Він підключається до карти через невеликі роз'єми у верхній частині карти і використовує спеціальний інтерфейс CSI, розроблений спеціально для камер. Сам датчик невеликий, всього 25 мм x 23 мм x 9 мм. Важить приблизно 3 г, ідеально підходить для мобільних додатків або інших невеликих програм, де важлива вага та розмір. Для підключення до плати Raspberry Pi використовується короткий стрічковий кабель.

- 8 мегапіксельна камера здатна робити фото з розширенням в 3280 x 2464 пікселя
- Запис відео з розширенням 1080p30, 720p60 та 640x480p90
- Все програмне забезпечення підтримується в останній версії Raspberry Operating System
- Застосування: кабельна камера відеоспостереження, виявлення руху, покадрові фотографії.

Модуль камери Raspberry Pi зазвичай поставляється з невеликим пластиковим колесом. Він призначений для регулювання фокусування об'єктива. Хоча заводські налаштування фокусування зазвичай ідеальні, якщо ви використовуєте камеру для

зніmkів дуже великим планом, ви можете посунути кільце на об'єктиві та обережно повернути його, щоб вручну налаштувати фокус на точку.

Як і будь-який аксесуар, модуль камери потрібно під'єднати або від'єднати від Raspberry Pi лише тоді, коли живлення вимкнено, а шнур живлення від'єднано. Якщо ваш Raspberry Pi увімкнено, виберіть «Вимкнути» в головному меню, зачекайте, поки він вимкнеться, а потім відключіть його.

### 2.2.3 Датчик руху PIR

Датчик PIR (піроелектричний інфрачервоний датчик) є пасивним пристроєм, оскільки він лише вловлює інфрачервоне випромінювання і не випромінює його. Він виявляє будь-яку зміну «нагрівання» і, виявивши це, встановлює вихідний контакт на HIGH.

Будь-який нагрітий предмет створює випромінювання в інфрачервоному діапазоні, і організм людини не є винятком з цього правила. PIR-датчик здатний виявити найменші зміни випромінювання в інфрачервоному діапазоні. При цьому він здатний виявляти не тільки нагріті тіла (людина, кішка, собака тощо), а й «холодні» (ненагріті) рухомі тіла, оскільки внаслідок тертя з повітрям рухоме тіло нагрівається і тому створює зміну інфрачервоного випромінювання, яке вловлюється PIR-датчиком.

Основним компонентом PIR-датчика є піроелектричний датчик, прямокутний кристал, показаний на наступному рисунку 2.4.

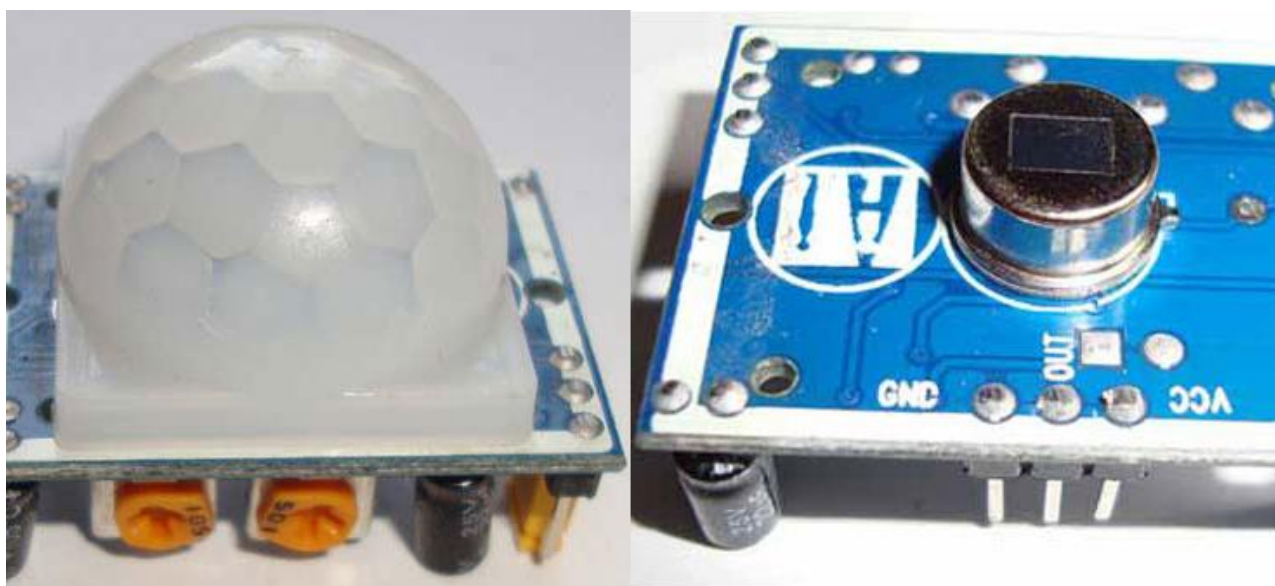


Рисунок 2.4 – Прямокутний кристал.



У конструкції зібраного датчика він закритий пластиковою кришкою. Датчик також містить ряд резисторів, конденсаторів та інших компонентів, необхідних для його роботи.

На рисунку 2.5 представлено зображення датчика руху PIR.

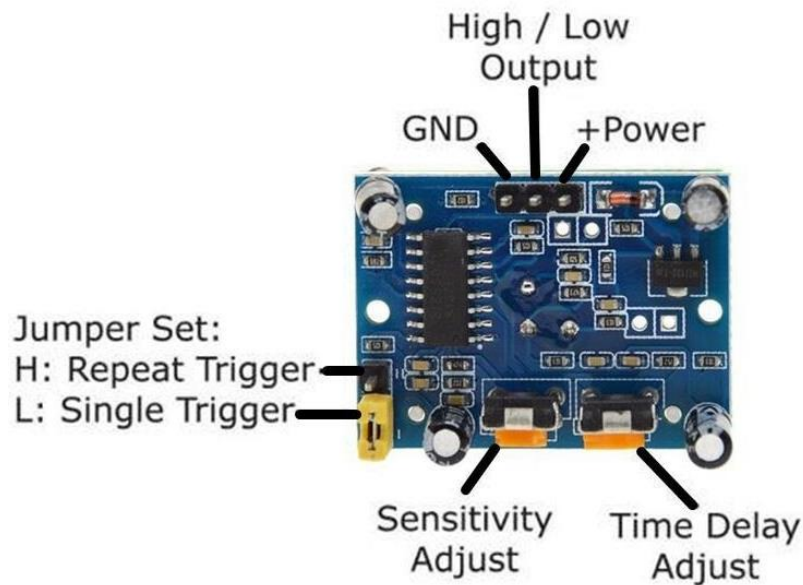


Рисунок 2.5 – Датчик руху PIR

Характеристика:

- відстань виявлення руху - 4...7 м
- ефективний кут - <math><120^\circ</math>
- Час затримки - 5...200
- напруга живлення - 4.5...20В
- вихідний сигнал - High/Low TTL (3.3 В)
- Довжина x Ширина x Висота - 33x25x24

## 2.3 Схема підключення

1. Підключити плоский стрічковий кабель до модуля камери. Рисунок 2.6

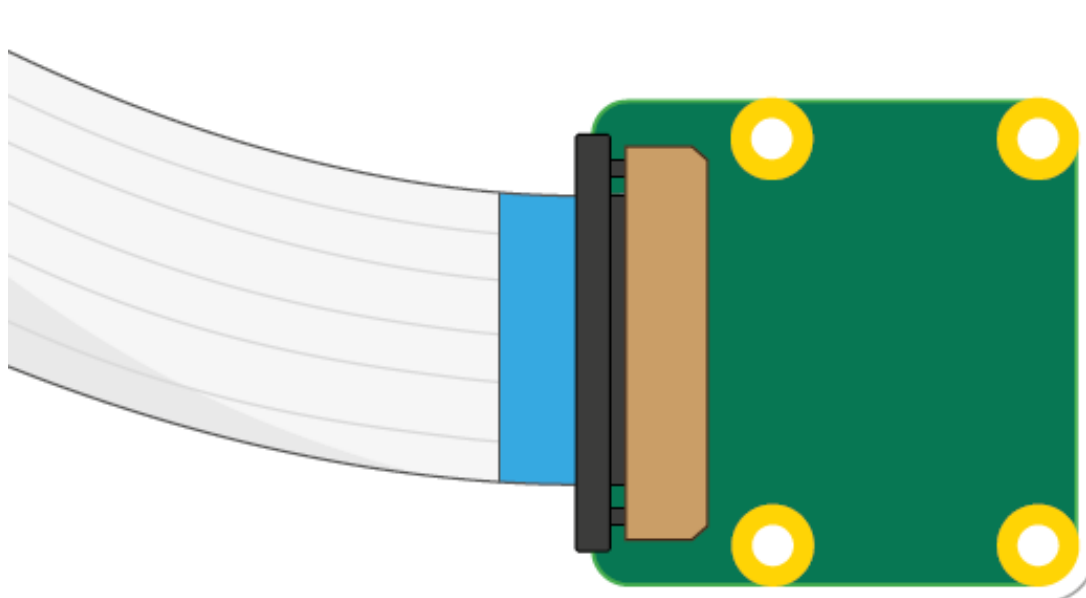


Рисунок 2.6 – Підключення кабелю до камери.

2. Підключити плоский стрічковий кабель до плати Raspberry Pi. Рисунок 2.7

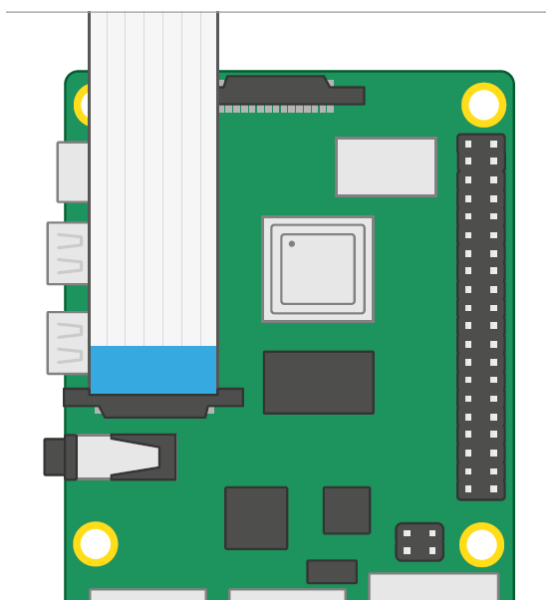


Рисунок 2.7 – Підключення кабелю до плати Raspberry Pi.

3. Загальний вигляд підключеної камери до мікрокомп'ютера Raspberry Pi.

Рисунок - 2.8



Рисунок 2.8 Підключення модуля камери до Raspberry Pi

## 2.4 Принцип роботи

Піроелектричний датчик руху складається з двох основних частин. Кожна з частин включає в себе спеціальний матеріал, чутливий до інфрачервоного випромінювання. В даному випадку лінзи особливо не впливають на роботу датчика, так що ми бачимо дві ділянки чутливості всього модуля. Коли датчик знаходиться в стані спокою, обидва сенсори визначають однакову кількість випромінювання. Рисунок 2.9 – Принцип роботи датчика

Наприклад, це може бути випромінювання приміщення або навколишнього середовища на вулиці. Коли теплокровний об'єкт (людина або тварина), проходить повз, він перетинає зону чутливості першого сенсора, в результаті чого на модулі PIR датчика генеруються два різних значення випромінювання. Коли людина залишає зону чутливості першого сенсора, значення вирівнюються. Саме зміни в показах двох датчиків реєструються і генерують імпульси HIGH або LOW на виході.

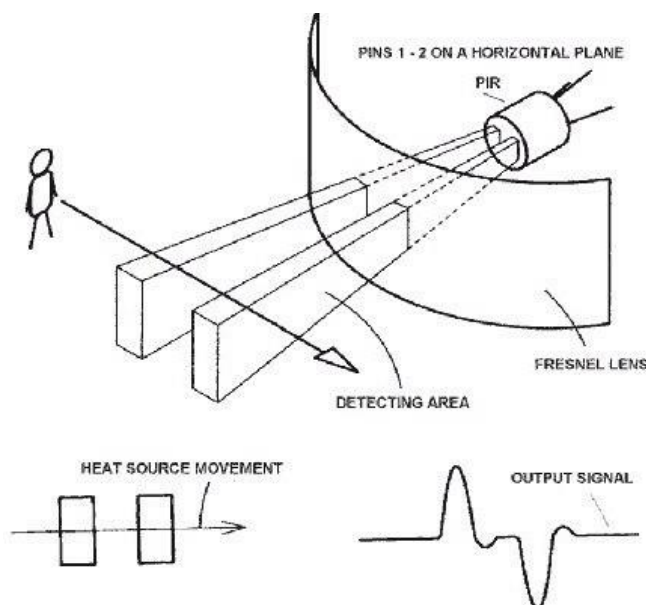


Рисунок 2.9 – Принцип роботи датчика.

## 2.5 Модуль ESP8266

Оновлений Wi-Fi модуль ESP8266 версії ESP-01S на базі мікросхеми ESP8266EX із вбудованим стеком протоколу TCP/IP та керуванням AT-командами. Чіп створений для використання в розумних розетках, mesh-мережах, IP-камерах, бездротових сенсорах, електроніці, що носить, і так далі. Рисунок 2.10 – ESP8266

Передбачено два варіанти використання чіпа: 1) міст UART-WIFI, коли модуль на базі ESP8266 підключається до існуючого рішення на базі будь-якого іншого мікроконтролера та керується AT-командами, забезпечуючи зв'язок рішення з інфраструктурою Wi-Fi; 2) реалізуючи нове рішення, що використовує сам чіп ESP8266 як керуючий мікроконтролер.

Мікроконтролер ESP8266 працює з зовнішньої flash-пам'яттю по інтерфейсу SPI. Її обсяг варіюється від 512 Кбайт до 4 Мбайт. При бажанні і вмінні мікросхему пам'яті можна буде перепаяти на версію до 32 Мбайт.

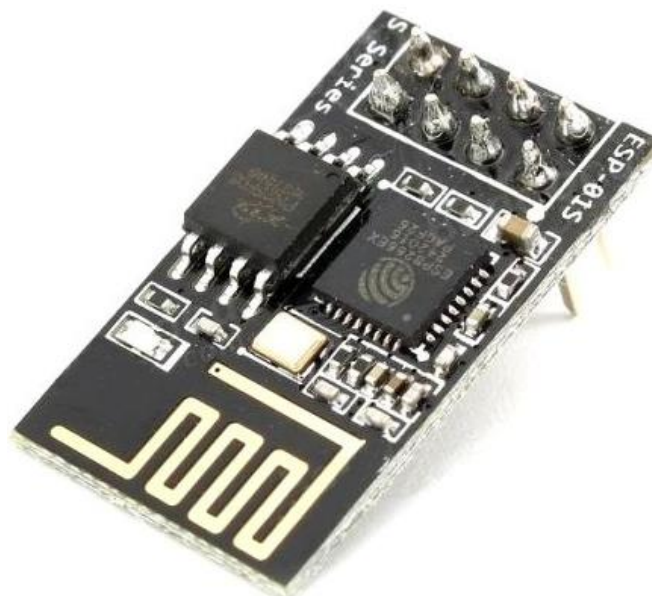


Рисунок 2.10 – ESP8266

### 2.5.1 Характеристики:

- підтримка WiFi протоколів 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- вбудований стек TCP/IP
- вбудований TR перемикач, balun, LNA, підсилювач потужності та відповідність мережі
- вбудований PLL, регулятори, та система управління живленням
- вихідна потужність +19.5 дБм у режимі 802.11b
- SDIO 2.0, SPI, UART
- STBC, 1x1 MIMO, 2x1 MIMO
- пробудження та відправлення пакетів: до 22 мс
- споживання у режимі Standby до 1.0 мВт (DTIM3)
- розміри: 24.5x14 мм

### 2.5.2 Контакти:

- Vcc - живлення, +3,3В (максимально 3,6В)
- GND - загальний
- TXD - передача даних (рівень 3,3В)
- RXD - прийом даних (рівень 3,3В)
- CH\_PD - вимкнення модуля (низький рівень активний, для включення модуля слід подати Vcc)
- GPIO0 - контакт загального призначення 0
- GPIO2 - контакт загального призначення 2 (LED blue, low-on)
- RST - скидання модуля (низький рівень активний)

### 2.5.3 Способи роботи

Є два способи роботи з ESP8266:

- управління через AT-команди і автономна робота зі своєї прошивкою. У першому випадку ESP8266 працює тільки в парі з іншими МК, у другому — може працювати незалежно (хоча ніхто не забороняє приєднати її до іншого МК).
- «3 коробки» МК поставляється з прошивкою для роботи через AT-команди. Для цього ESP8266 підключається до будь-якого іншого МК по UART-інтерфейсу. Для демонстрації роботи AT-команд ESP8266 можна підключити до комп'ютера через USB-UART перехідник і запустити монітор послідовного порту (наприклад з Arduino IDE).

### 2.5.4 Програмування ESP8266

У більшості випадків набагато зручніше прошивати МК і працювати з ним зі своєю програмою. Однак тут є деякі особливості:

- «Чистий» ESP8266, наприклад ESP-01. Тут знадобиться USB-UART перехідник, який потрібно підключити до МК. Цей перехідник обов'язково повинен бути на 3-вольтову логікою, оскільки інакше можна легко пошкодити МК.
- Другий випадок ідентичний першому, окрім того, що замість перехідника можна використовувати будь-яку плату Arduino. Досить спеціальним чином підключити ESP8266 до UART-контактів Arduino, а її саму «відключити», замкнувши контакт апаратного RESET на землю. Звичайно, живити ESP8266 потрібно буде від шини живлення 3.3 В. В цьому випадку в якості перехідника USB-UART буде виступати міст (найчастіше CH340) на самій платі Arduino.
- Кращий варіант — це плата з USB-UART мостом на борту (наприклад NodeMCU, WeMos та інші). В цьому випадку нічого додатково робити не потрібно, достатньо підключити плату через USB.

## 2.6 Модуль ESP32-CAM та потокова передача відео

Модуль камери на базі плати ESP-32 має вбудований Wi-Fi модуль та дозволяє у передавати відео потоковому режимі. При цьому потрібно враховувати такі особливості потокової передачі даних:

1. Високошвидкісна передачі захоплених кадрів зображення — залежить від використовуваного стандарту.
2. Асинхронність передачі даних для досягнення низької затримки.
3. Відображення захоплених даних у режимі FIFO.
4. Швидке оновлення пікселів у кадрах для уникнення мерехтіння

### 2.6.1 Основні характеристики ESP32-CAM

Бездротовий модуль - ESP32-S WiFi 802.11 b/g/n має такі характеристики:

1. Зовнішнє сховище - слот для карт мікро - SD ємністю до 4 ГБ;
2. Формат зображення - JPEG (тільки OV2640), BMP, відтінки сірого;
3. Світлодіодний спалах.
4. Контакти – 16 з інтерфейсами UART, SPI, I2C, PWM
5. Напруга живлення - 5 В;
6. Величина струму:
  - спалах вимкнено/увімкнено - 180/ 310 мА;
  - глибокий сон - 6 мА;
  - модем-сон - 20 мА;
  - легкий сон - 6,7 мА.
8. Розміри - 40,5 x 27 x 4,5 мм

#### На платі присутні:

- Бездротовий модуль ESP32-S з інтегрованим Wi-Fi і Bluetooth контролерами.
- Камера OV2640 - йде окремо, але на платі передбачений роз'єм для підключення.
- Роз'єм для карток пам'яті мікро-SD.
- Світлодіод, який відіграє роль спалаху.
- 32-розрядний процесор з тактовою частотою 160 або 240 МГц.
- Wi-Fi: 802.11 b/g/n
- Bluetooth: v4.2 BR/EDR and BLE
- Велика кількість периферійних модулів, у тому числі:
  - SPI x 4
  - ADC
  - DAC x 2
  - UART x 3
  - CAN
  - I2C x 2
- 520 КБ SRA



Рисунок 2.11 – ESP32-CAM

### 2.6.2 Алгоритм потокової передачі відео з використанням WebSocket

Особливістю протоколу WebSocket є:

1. Можливість передавати дані в режимі реального часу з низькими накладними витратами.
2. Клієнту або серверу не потрібно чекати один одного для початку або завершення передачі даних.
3. Підтримка як текстової, так і двійкової передачі даних. Це допомагає в обміні користувацькими командами і двійковими зображеннями.

Для того, щоб передавати зображення з камери у браузер по протоколу WebSocket потрібно дотримуватися наступного алгоритму:

1. Встановити з'єднання по WebSocket між браузером і ESP32.
2. Браузер відправляє повідомлення "start" у ESP32.
  - а. Початкове повідомлення містить роздільну здатність зображення: 80x60, 160x120 або 320x240.
3. ESP32 починає захоплення кадрів і відправляє їх у браузері за допомогою WebSocket.sendBIN. Формат зображення RGB565. Отже, загальний розмір кадру дорівнює:
 
$$\text{size} = w * h * 2;$$
4. Пам'яті потрібно виділити стільки, щоб поміститися в кадр розміром 160x120x2 байта (QQVGA).
5. При роздільній здатності 320x240 кадр знімається 2 рази. При першому захоплення відправляється перша половина кадру, а при другому захопленні інша



половина відправляється через веб-сокет. Прапорці *start* і прапор *end* використовуються для інформування браузера про часткове порядку кадрів.

6. Після отримання прапорця *end* браузер запитує в ESP32 наступний кадр. І далі продовжуємо роботу з кроку 3.

### 2.6.3 Керуюча програма для отримання потокового відео з ESP32-CAM

Для швидкого запуску ESP32-CAM використовуємо існуючі приклади та бібліотеки для Arduino IDE

**Крок 0.** Інсталяція Arduino IDE - завантажуюмо з офіційного сайту, встановлюємо і запускаємо.

**Крок 1.** У свіжовстановленій IDE необхідно інтегрувати підтримку модуля ESP32. Йдемо в настройки (File - Preferences) і в нижній частині вікна налаштувань в Additional Boards Manager URLs додаємо рядок:

```
https://dl.espressif.com/dl/package_esp32_index.json
```

В результаті отримуємо наступне Рисунок 2.12:

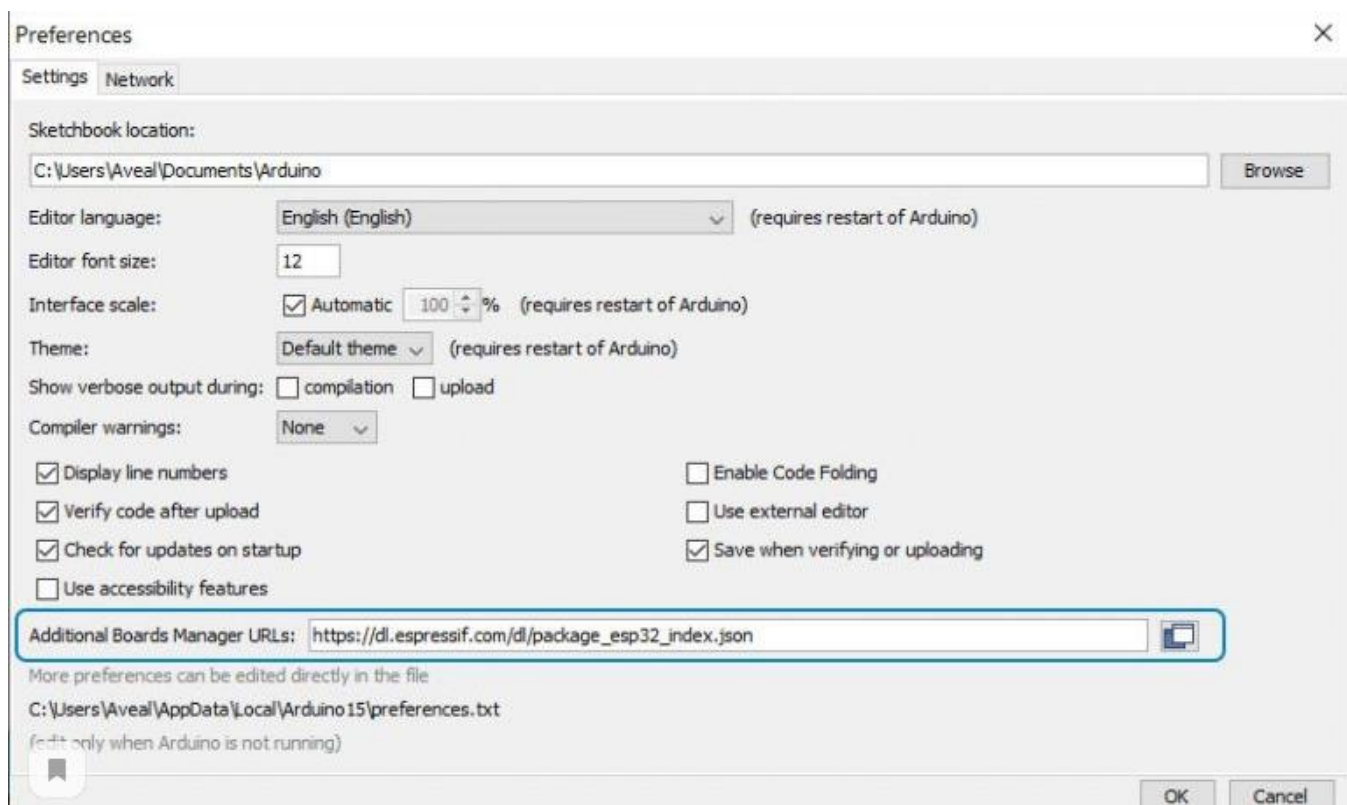


Рисунок 2.12 – Налаштування

**Примітка.** Якщо у цьому рядку вже збережене яке-небудь посилання, то можна без проблем додати ще одне, розділивши їх комою:

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json),  
<https://link2.com/dl/file2.json>,  
<https://link3.com/dl/file3.json>

**Крок 2.** Переходимо в Boards Manager: Tools - Board - Boards Manager (рис. 2.21):

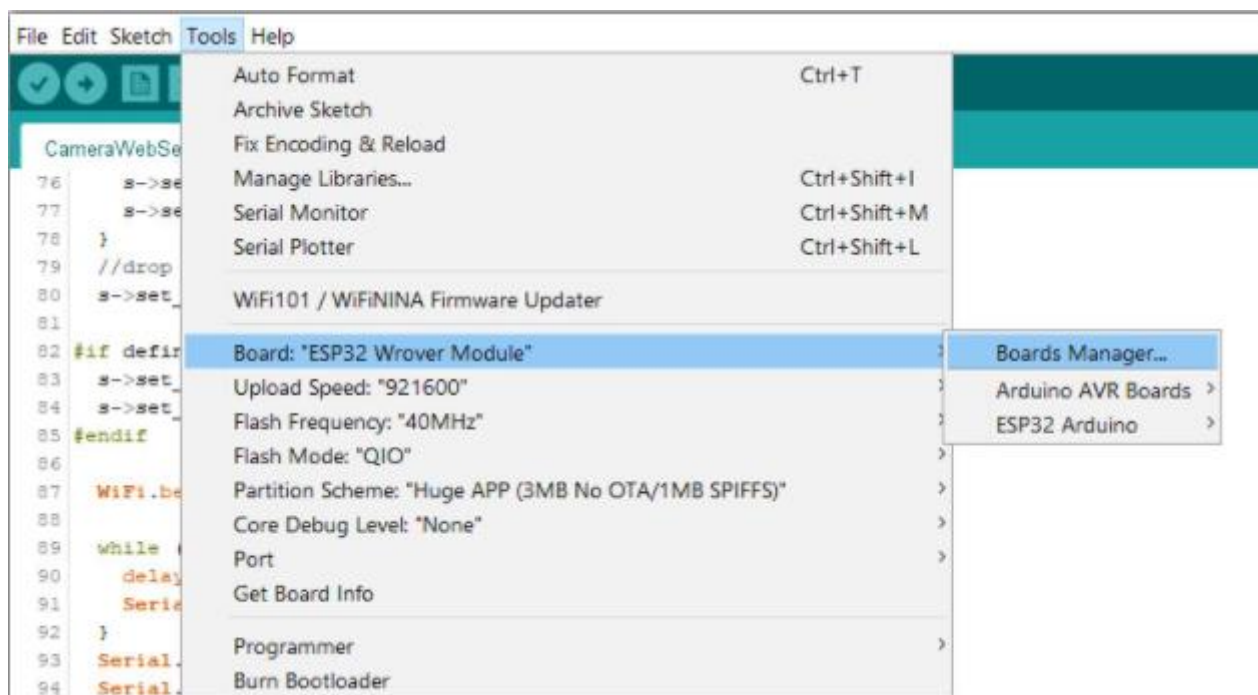


Рисунок 2.13 – Перехід в меню Boards Manager

У вікні пошуку вводимо "esp32", отримуємо один варіант і тиснемо Install  
Рисунок 2.14:

Кафедра комп'ютерної інженерії  
Система детектування руху на основі ESP8266-CAM та Raspberry Pi

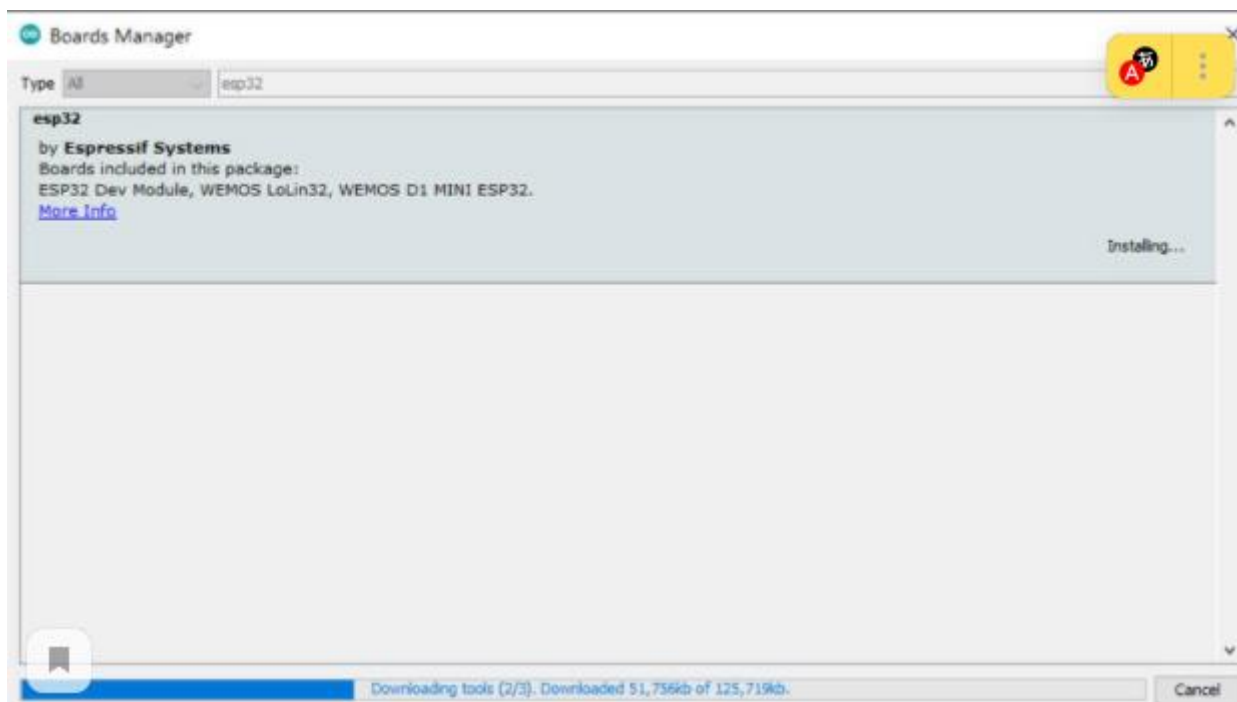


Рисунок 2.14 – Встановлення

Очікуємо завершення операції і переходимо до наступного етапу.

**Крок 3.** Відкриваємо Arduino IDE приклад для роботи з камерою. Для цього йдемо в File - Examples - ESP32 - Camera – CameraWebServer Рисунок 2.15:

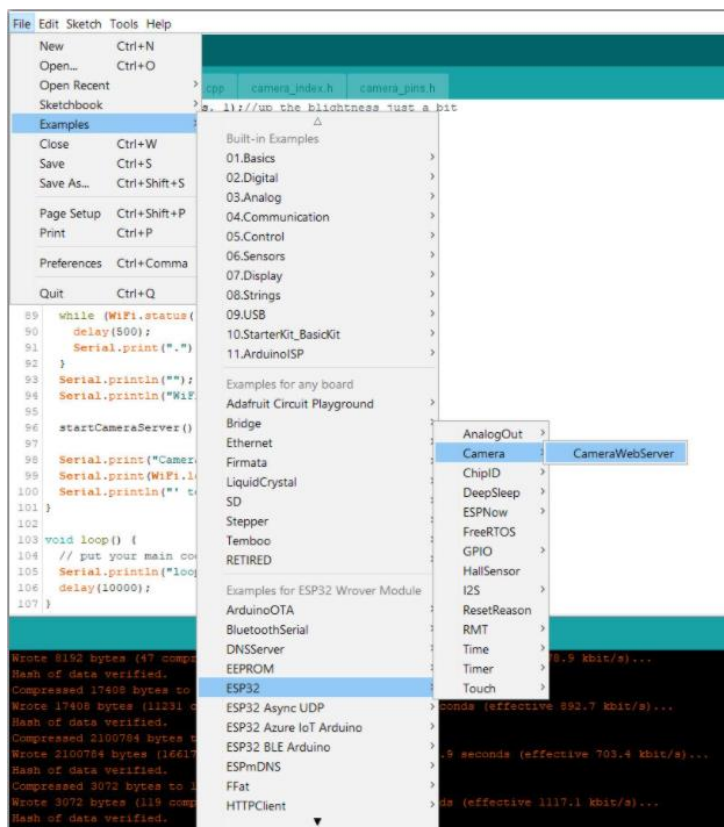


Рисунок 2.15 – Налаштування камери

Отримуємо готовий проект з ініціалізацією в опції *setup()* і основним циклом програми у функції *loop()*.

**Крок 4.** Тепер необхідно трохи модифікувати код керуючої програми для того, щоб плата і камера успішно запрацювали. Насамперед вибираємо модель камери. Здійснюється це коментуванням непотрібного і розкоментуванням потрібного:

1. `// Select camera model`
2. `//#define CAMERA_MODEL_WROVER_KIT`
3. `//#define CAMERA_MODEL_ESP_EYE`
4. `//#define CAMERA_MODEL_M5STACK_PSRAM`
5. `//#define CAMERA_MODEL_M5STACK_WIDE`
6. **`#define CAMERA_MODEL_AI_THINKER`**

Друга зміна полягає в установці імені мережі і пароль для підключення по Wi-Fi. Наприклад:

1. `const char* ssid = "home";`
2. `const char* password = "pass1234";`

**Крок 5.** У меню Tools нам потрібно задати актуальні параметри для програмування плати - вибрати плату, номер COM-порту і т. д. (рис. 2.16):

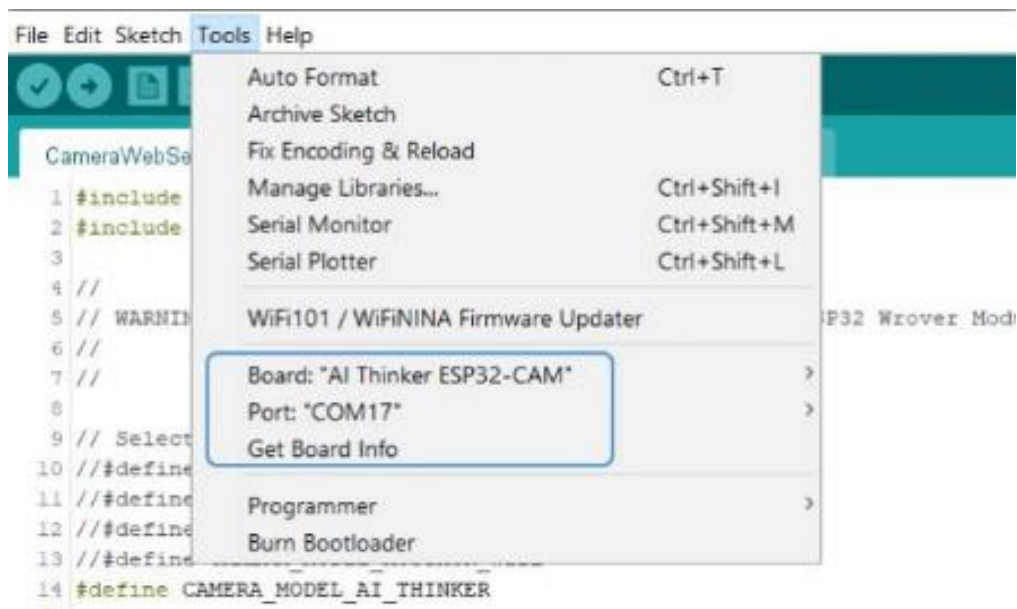


Рисунок 2.16 – Список налаштувань

**Крок 6.** Завантажуємо код програми до плати, для цього натискаємо кнопку *upload*, не забувши попередньо підтягнути пін GPIO0 до землі (рис. 2.17):

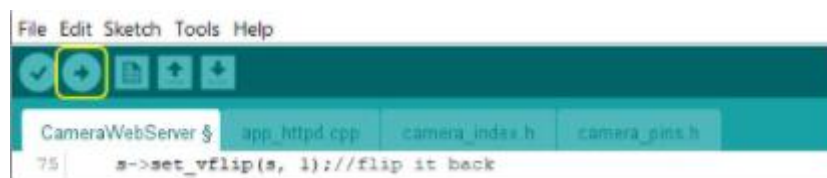


Рисунок 2.17 – Прошивання плати

Завершальна частина успішного лога прошивки виглядає приблизно так:

```
Compressed 3072 bytes to 119...
Wrote 3072 bytes (119 compressed) at 0x00008000 in 0.0 seconds (effective
1068.5 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
```

**Крок 7.** Відключаємо підтяжку GPIO0 і запускаємо Serial Monitor (Tools - Serial monitor) (рис. 2.18):

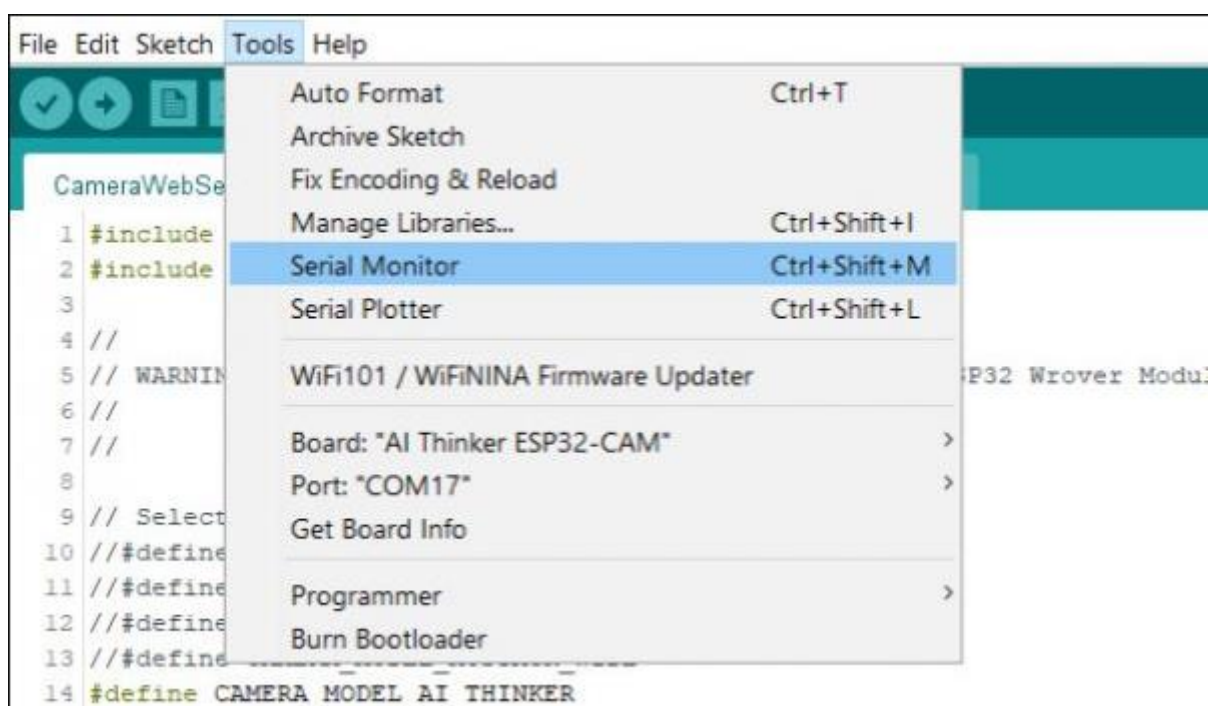


Рисунок 2.18 – Відключення

Тепер можемо перезапустити плату кнопкою Reset і на екрані з'являться рядки, які сигналізують про успішне підключення до мережі Wi-Fi :

```
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.0.104' to connect
```

**Крок 8.** Переходимо у браузер за вказаною IP-адресою, щоб наочно побачити результат роботи програми. Далі знаходимо внизу кнопку Start Stream і отримуємо зображення з OV2640 (рис. 2.19):

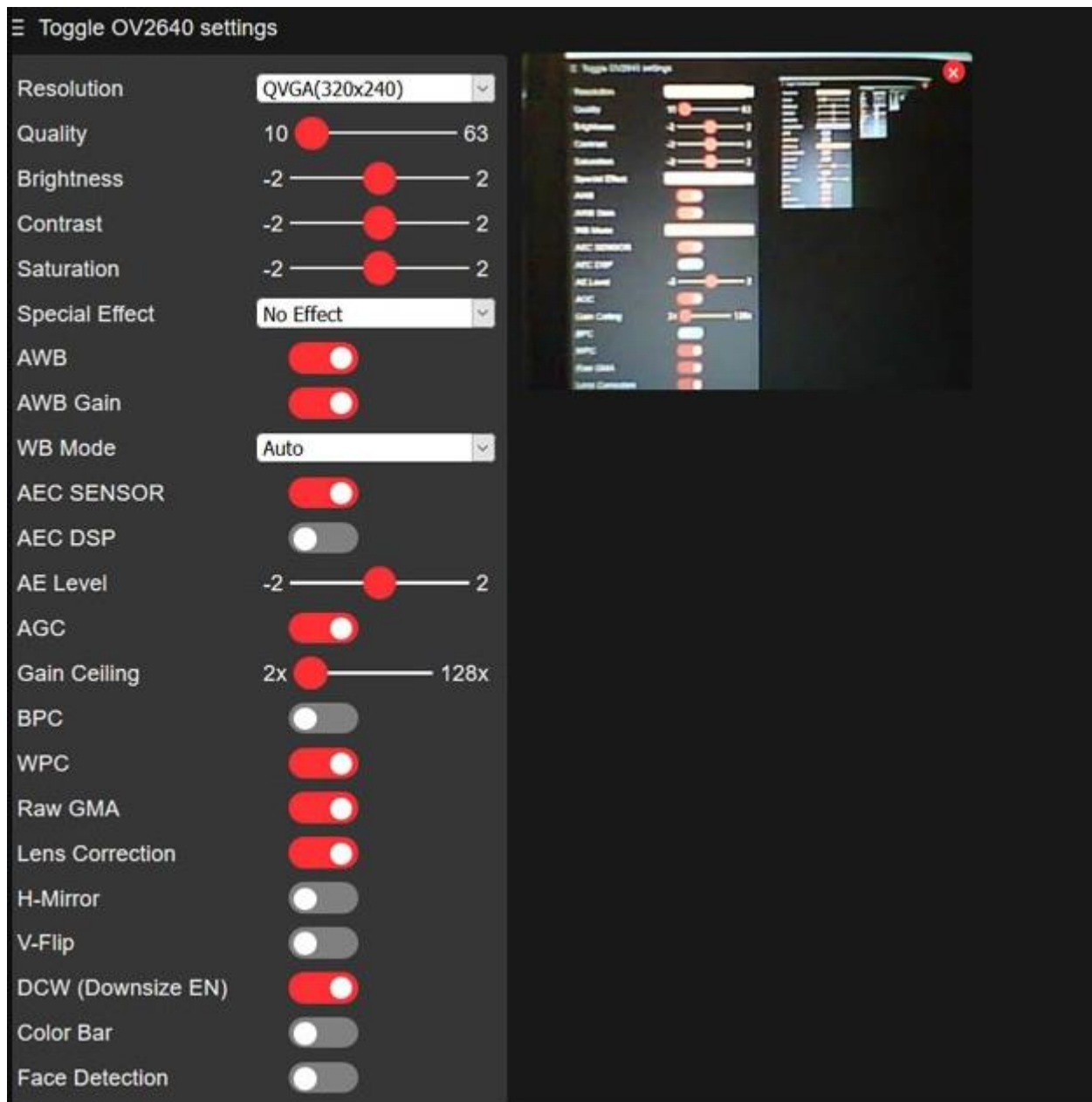


Рисунок 2.19 – Отримане зображення з OV2640

Одними із корисних режимів камери є режим розпізнавання людини. Для цього у налаштуваннях вмикаємо *Face Detection* і *Face Recognition* і далі кнопкою *Enroll Face* реєструємо людину. Після цього камера визначить, чи знаходиться в кадрі зареєстрована особа або ні.

### **Висновок до другого розділу**

У другому розділі було розглянуто пристрій системи детектування руху, та наведена блок-схема візуальних компонентів, які створюють функціональний і в одночас простий та зрозумілий інтерфейс. Описано використані елементи для виготовлення робочої системи детектування, та розглянули їх характеристики. Розібрано та сконструювали схему детектування руху. Створили блок-схеми програми.

## **3 ПРОГРАМНА ЧАСТИНА**

### **3.1 Опис алгоритму роботи:**

1. При подачі живлення, пристрій стає активним (на мікрокомп'юторі горить індикатор живлення, та блимає індикатор загрузки ОС).
2. Якщо одноразово натиснути на кнопку «А», то камера почне працювати якщо є рух перед датчикм руху (при повторному натисканні кнопки камера перестає реагувати на датчик).
3. У робочому режимі пристрій під час руху перед датчиком буде записувати відео на камеру, та зберігатся на флеш карту, яка підключена до мікрокомп'ютера.
4. В робочому режимі в будь який момент можна переглянути переглянути пряму трансляцію с камери завдяки підключенню до інтернет мережі.
5. В режимі очікування пристрій пристрій не веде запис на флеш карту, так как очікує сигнал від датчику руху.

### **3.2 Raspberry Pi перший запуск**

Для першого запуску нам потрібно:

- Raspberry Pi
- USB Клавіатура
- USB мишка
- SD/MMC/SDIO карта пам'яті з об'ємом від 2 до 32 Гб
- Micro USB адаптер для живлення Raspberry Pi
- Монітор з HDMI входом, або телевізор, або TV тюнер з RCA роз'ємом
- HDMI кабель для підключення до монітору або RCA кабель для підключення до телевізору або тюнеру
- Програма Win32DiskImager для запису образу на флешку
- Образ ОС Raspbian «wheezy»
- Картридер для считування SD/MMC/SDIO карт пам'яті



### 3.3 Запис образу **Raspbian** на карту пам'яті.

Для запису ОС на флешку нам потрібно скачати ОС та програму Win32DiskImager. Після встановлення програми та скачування ОС потрібно:

- Вставити флеш карту в картридер та підключити до ПК
- Запустити програму Win32DiskImager
- В полі потрібно указати місце знаходження скачаної ОС
- Обрати в меню «Device» букву диску з вашою флешкою
- Натиснути кнопку Write та дочекатись закінчення запису як на Рисунок - 3.1

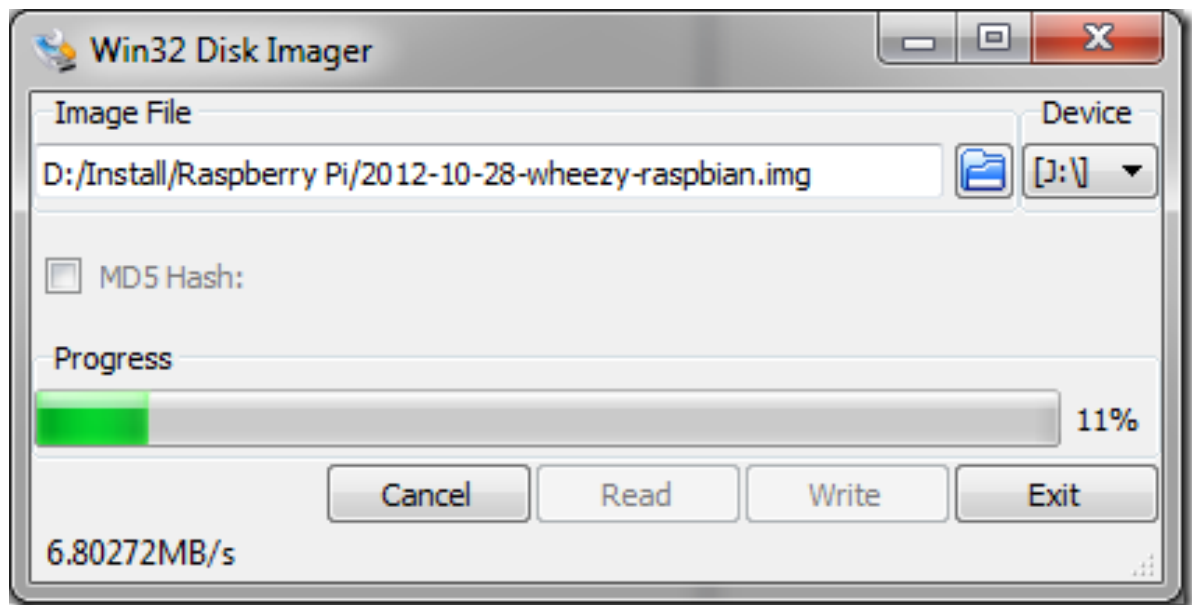


Рисунок 3.1 – Завантаження ОС на флеш карту

### 3.4 Підключення **Raspberry Pi**

- Підключіть карту пам'яті до Raspberry Pi
- Підключіть USB-клавіатуру (і USB-мишу, якщо потрібно)
- Під'єднайте кабель HDMI до монітора на одному кінці, а кабель Raspberry Pi або RCA до телевізора (телевізійного тюнера) і Raspberry Pi до іншого кінця (в RCA буде використовуватися лише жовтий роз'єм)
- Увімкніть монітор або телевізор (ТВ-тюнер)
- Підключіть блок живлення mini USB до Raspberry Pi
- Підключіть адаптер живлення Mini USB до розетки

### 3.5 Налаштування **Raspberry Pi**

При першому запуску Raspberry на екрані з'являється меню налаштувань.  
Рисунок - 3.2



Кафедра комп'ютерної інженерії  
Система детектування руху на основі ESP8266-CAM та Raspberry Pi

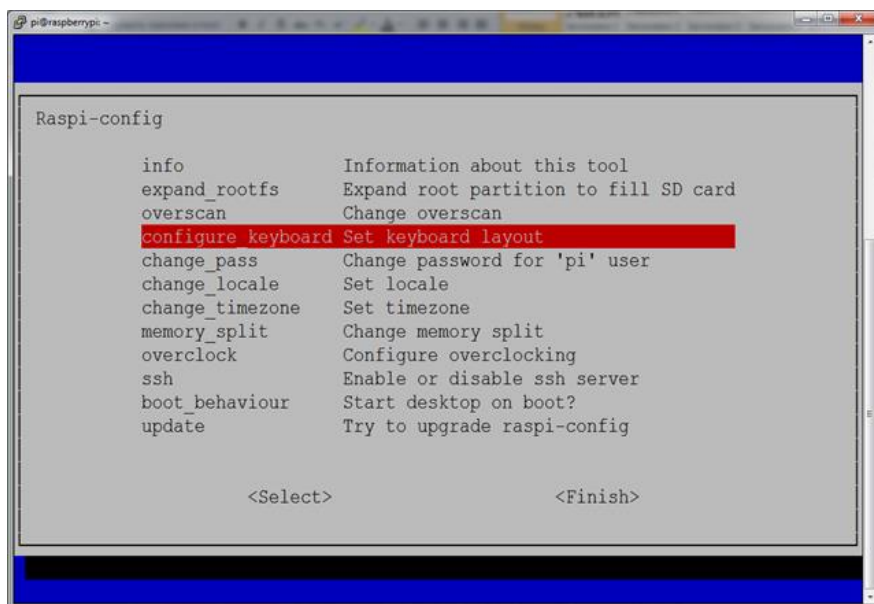


Рисунок 3.4 – Налаштування драйвера.

В графі change\_pass змінюємо пароль користувача на зручний. Пароль не відображається та його потрібно повторити. Рисунок – 3.5

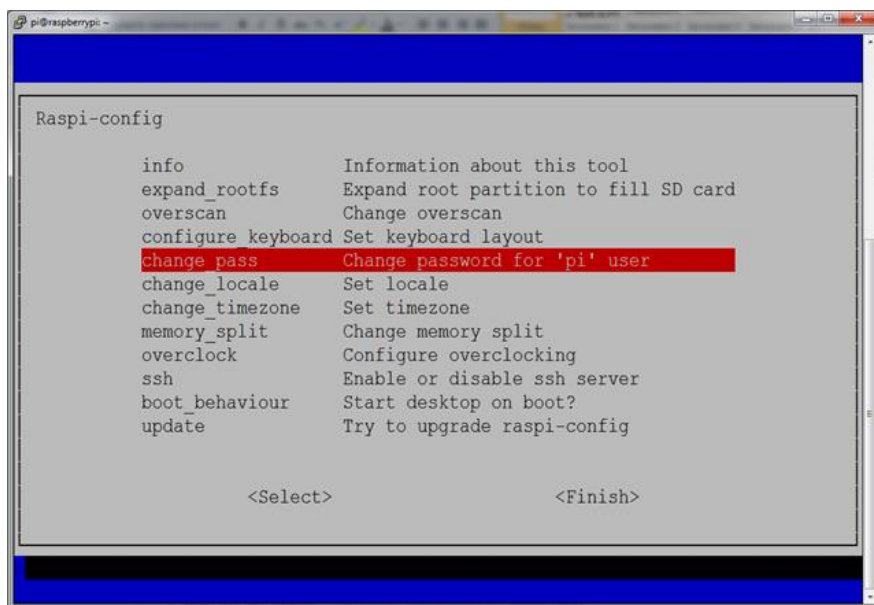


Рисунок 3.5 – Налаштування паролю.

Функція change\_locate потрібна для налаштування мови та кодування.  
Рисунок – 3.6

Кафедра комп'ютерної інженерії  
Система детектування руху на основі ESP8266-CAM та Raspberry Pi

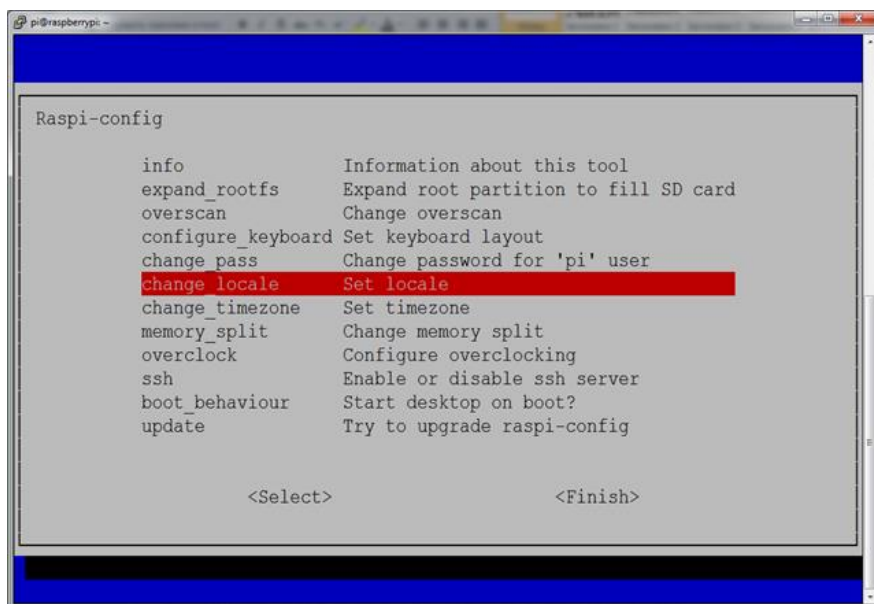


Рисунок 3.6 – Налаштування мови.

В графі change\_timezone оберемо часовий пояс. В Raspberry немає свого годиннику, тому час береться з інтернету. Для зручності рекомендується обрати свій регіон. Рисунок – 3.7

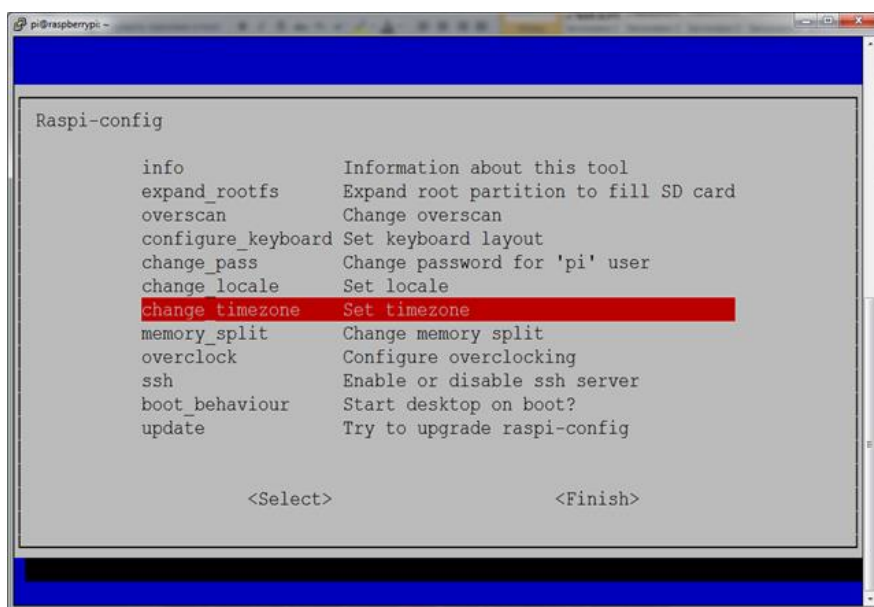


Рисунок 3.7 – Налаштування часового поясу.

Memory\_split – розподіляє пам'ять Raspberry Pi. Рисунок – 3.8.

Кафедра комп'ютерної інженерії  
Система детектування руху на основі ESP8266-CAM та Raspberry Pi

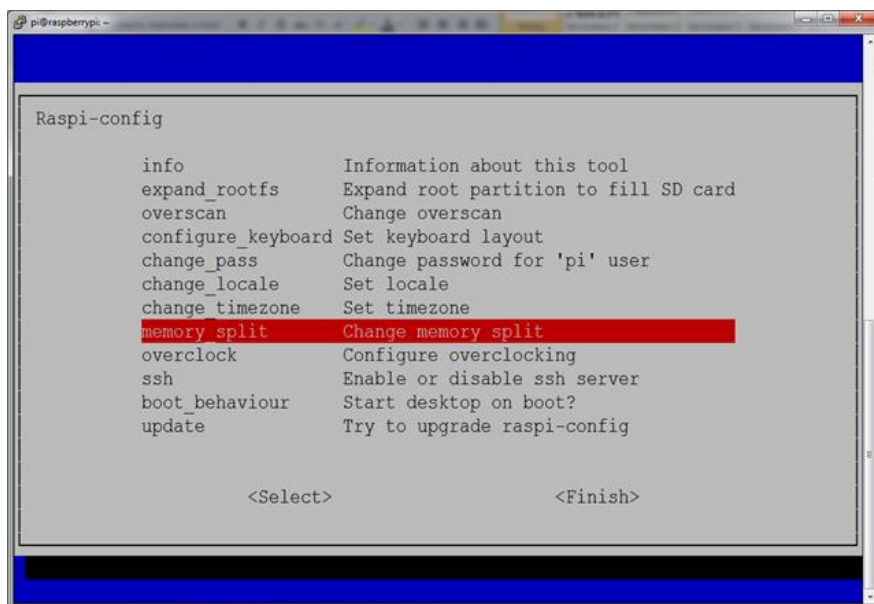


Рисунок 3.8 – Налаштування розподілу пам'ятю.

Для роботи в консолі буде достатньо 16Мб а для перегляду відео доведеться пожертвувати від 64 до 128 Мб. Рисунок – 3.9 налаштування ОС. Для оперативної пам'яті нам дається на вибір декілька варіантів: 16, 34, 64, 128 або 256.

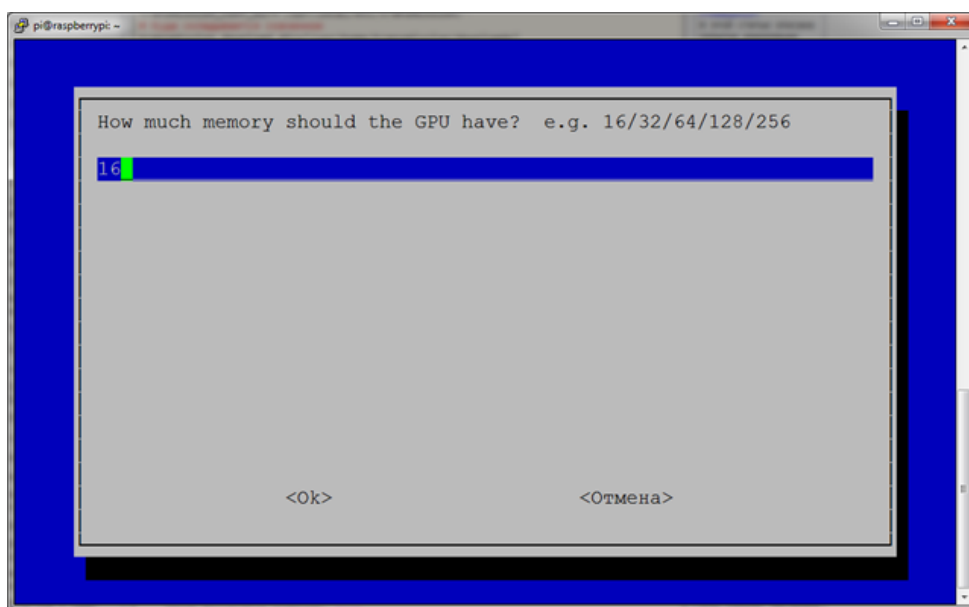


Рисунок 3.9 – Налаштування операційної пам'яті.

Функція overclock потрібна для розгону процесора Raspberry Pi. Рисунок – 3.10

Кафедра комп'ютерної інженерії  
Система детектування руху на основі ESP8266-CAM та Raspberry Pi

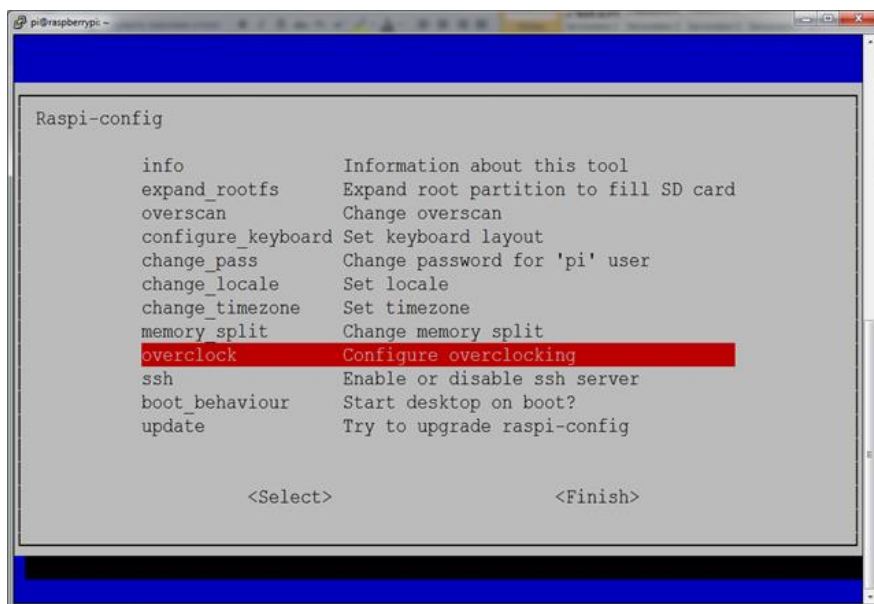


Рисунок 3.10 – Налаштування розгону процесору.

Для початку можна залишити значення по замовчуванню, а якщо необхідно буде обрати підходящий варіант, ви можете в будь який момент змінити на необхідний. Рисунок – 3.11

Але потрібно урахувати той факт що при збільшенні частоти, також збільшується енерго затрати.

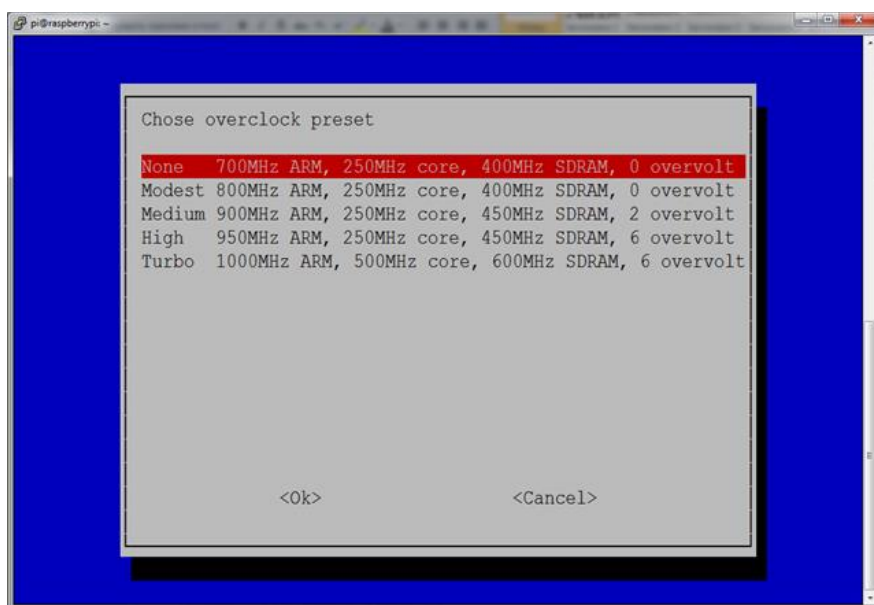


Рисунок 3.11 – Налаштування частоти процесору.

Функція ssh потрібна для налаштування віддаленого управління. Рисунок – 3.12. Так як я планую використовувати перегляд даних з камери яка підключена до Raspberry Pi то, потрібно увімкнути SSH сервер.





Рисунок 3.14 – Завершення налаштування.

### 3.6 Корисні консольні програми для Raspberry Pi

Диспетчер завдань. Рисунок – 3.15

Для запуску програми використовується команда «top».

The image shows a terminal window titled 'pi@raspberrypi: ~' displaying the output of the 'top' command. The output includes system statistics and a table of running processes.

```

top - 13:32:51 up 18:08, 1 user, load average: 2,05, 2,33, 1,83
Tasks: 73 total, 1 running, 72 sleeping, 0 stopped, 0 zombie
%Cpu(s): 20,4 us, 2,3 sy, 0,0 ni, 63,9 id, 12,4 wa, 0,0 hi, 1,0 si, 0,0 st
KiB Mem: 189100 total, 175640 used, 13460 free, 2960 buffers
KiB Swap: 102396 total, 22272 used, 80124 free, 56084 cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 23225 eggdrop  20   0 13380 3468 1304  S  15,8  1,8   1:43.20 eggdrop
 2516 debian-t 20   0 46588 13m 1244  S   4,6  7,3  38:55.79 transmission-da
 7958 pi       20   0 6348 1368 1040  R   1,0  0,7   0:00.14 top
   32 root    20   0   0     0     0   D   0,7  0,0   30:01.47 mmcqd/0
   33 root    20   0   0     0     0   S   0,3  0,0   0:03.88 jbd2/mmcblk0p2-
  355 root    20   0   0     0     0   S   0,3  0,0   0:09.13 flush-179:0
 1722 root    20   0 32044  660  100  S   0,3  0,3   0:06.63 php5-fpm
 2252 mysql   20   0 309m  22m  196  S   0,3 12,0   4:21.85 mysqld
 3202 root    20   0 19936  748  360  S   0,3  0,4   0:01.43 smbd
 3623 pi      20   0 8640  944  520  S   0,3  0,5   0:33.93 eggdrop
    1 root    20   0  2136   88   68  S   0,0  0,0   0:03.91 init
    2 root    20   0   0     0     0   S   0,0  0,0   0:00.04 kthreadd
    3 root    20   0   0     0     0   S   0,0  0,0   0:00.32 ksoftirqd/0
    6 root    0 -20   0     0     0   S   0,0  0,0   0:00.00 khelper
    7 root    20   0   0     0     0   S   0,0  0,0   0:00.01 kdevtmpfs
    8 root    0 -20   0     0     0   S   0,0  0,0   0:00.00 netns
    9 root    20   0   0     0     0   S   0,0  0,0   0:00.58 sync_supers
  
```

Рисунок 3.15 – Диспетчер завдань.

Файловий менеджер Мс. Рисунок – 3.16

Команда для завантаження: `sudo apt-get install mc`

Команда для запуску: `Мс`



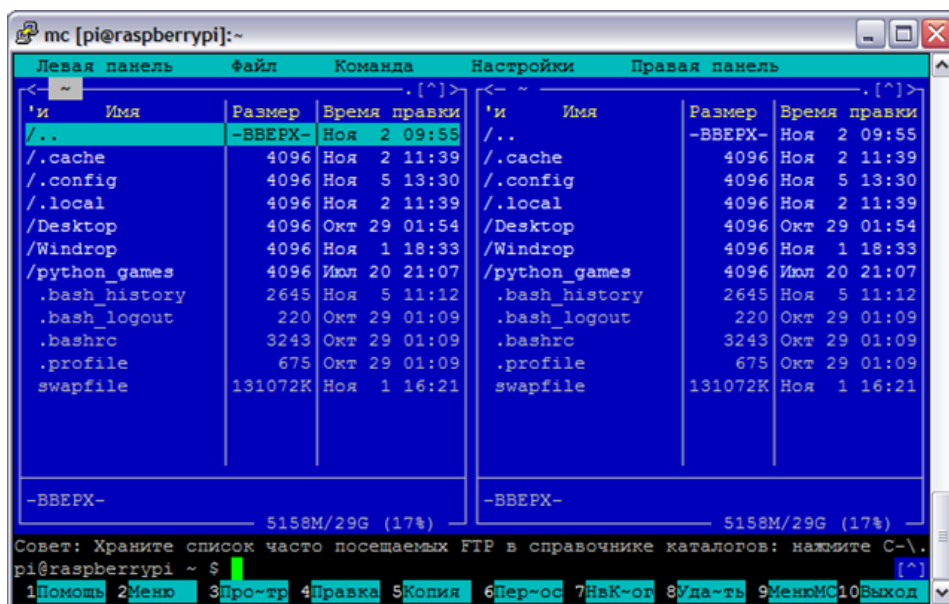


Рисунок 3.16 – Файловий менеджер

WeeChat – IRC клієнт. Рисунок – 3.17

Команда для завантаження: `sudo apt-get install weechat`

Команда для запуску: `weechat-curses`

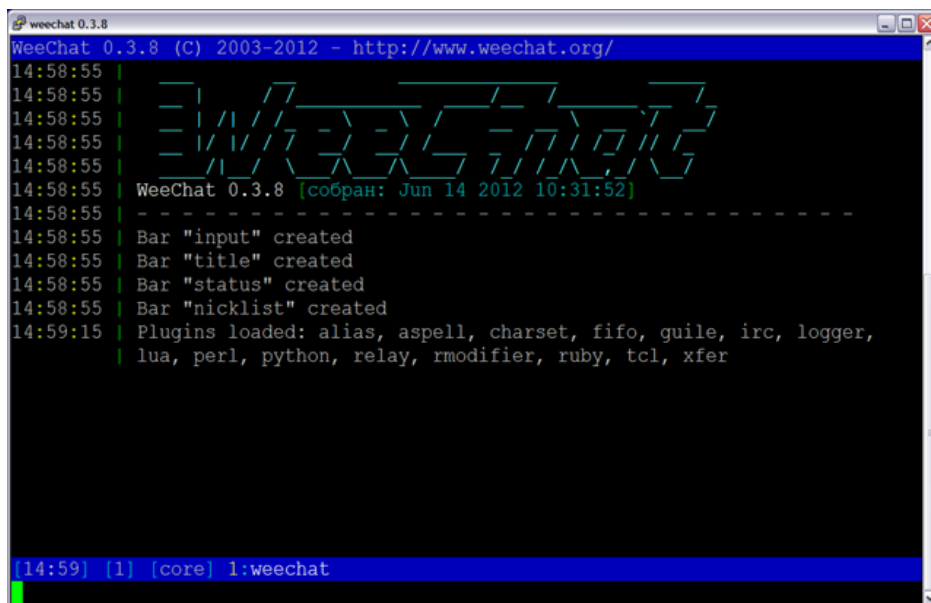


Рисунок 3.17 – WeeChat.

Links – текстовий браузер. Рисунок – 3.18.

Команда для завантаження: `sudo apt-get install links`

Команда для запуску: `links`

Клавіша Esc відкриє меню браузера.

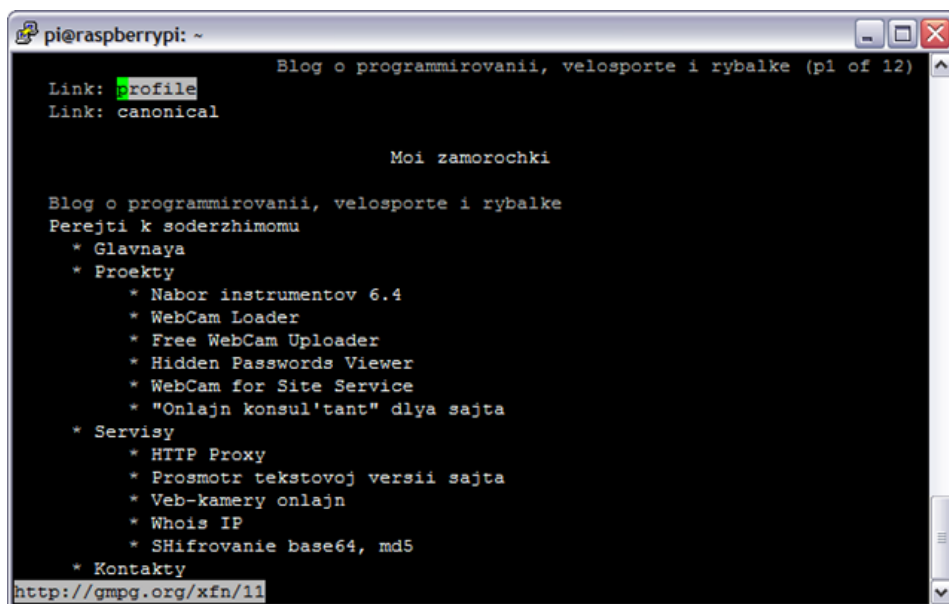


Рисунок 3.18 – Текстовий браузер.

### 3.7 Команди Linux

Найбільш популярні для використання консольні команди для ОС Linux

- **sudo** – ставиться перед командою, для виконання команди з правами root.
- **sudo reboot** – перезагрузка.
- **sudo halt** – вимкнення.
- **Ctrl+C** – вихід з відкритої консольної команди.
- **Shift+Ins** – вставити текст в консоль.
- **Ctrl+Ins** – копіювання виділеного тексту з консолі.
- **cd** – перехід в потрібну папку. Наприклад `cd /home/pi`.
- **dir** – відобразить вміст поточної папки.
- **pwd** – відобразить ваше поточне місцезнаходження.
- **date** – відобразить час та дату.
- **cal** – відобразить календар на поточний місяць.
- **cal -y** - відобразить календар на поточний рік.
- **wget** – завантажити файл в поточну папку.
- **sudo apt-get update** - оновить список пакетів з репозиторія.
- **sudo apt-get upgrade** – оновить встановлених пакетів.
- **sudo apt-get install** – встановить потрібний пакет. `sudo apt-get install mc`
- **стрілки верх та вниз** – перегортають раніше набрані команди.

### 3.8 Резервне копіювання Raspberry Pi

Тепер у нас є мініпально налаштована робоча система з невеликим набором програм. Щоб у випадку будь яких обставин не втратити всі налаштування, потрібно зробити backup даних.

- Вимкнути Raspberry Pi – в консолі пропишемо: `sudo halt`.
- Після вимкнення Raspberry, потрібно відключити її від живлення та дістати карту пам'яті.
- Вставити флеш карту в картрідер комп'ютера.
- Запустити програму Win32DiskImager.
- Вказати в меню Image File шлях, по якому буде збережений налаштований образ Raspbian.
- Вибрати в меню Device букву диска з вашою флешкою.
- Натиснути кнопку Read та дочекатися закінчення створення образу карти пам'яті.
- Закрити програму Win32DiskImager.
- Коректно вимкніть флеш карту.
- Виймнути карту пам'яті з картрідера.

### 3.9 Тестування та налаштування модуля камери.

Для того щоб впевнитись що камера встановлена та увімкнений інтерфейс в Raspberry Pi Configuration Tool, використаємо інструмент raspistill. Ця команда передозначена для захвата зображення з камери за допомогою інтерфейса командної строки (CLI, command-line interface) Raspberry Pi.

Для тесту камери потрібно ввести команду в термінал. Рисунок 3 – 19.

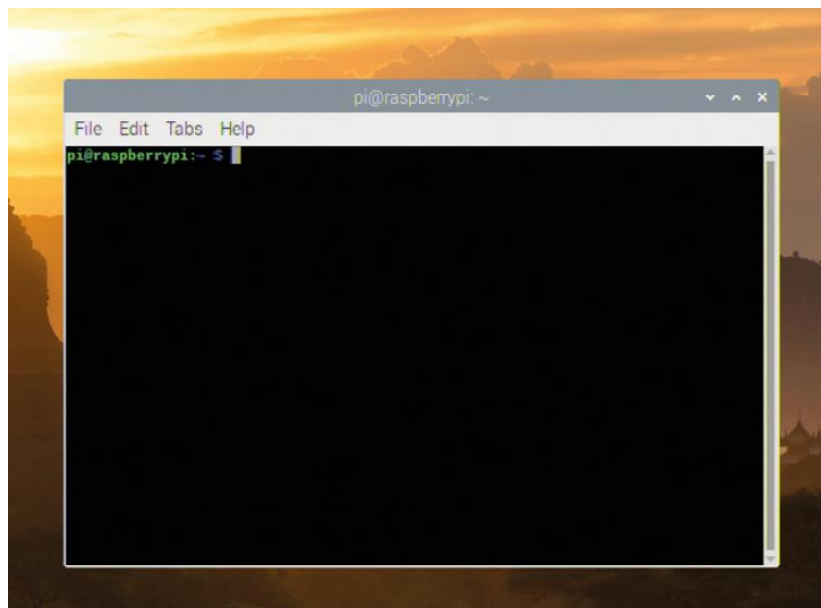


Рисунок 3.19 – Тест камери.

### **raspistill -o test.jpg**

Як тільки буде натиснуто клавішу ENTER, на екрані з'явиться велике зображення того що баче камера. Це називається попередній перегляд в реальному часі (**live preview**), якщо не буде нічого іншого вказано, то відео буде тривати 5 секунд. Як тільки закінчиться час, камера зробить одне зображення та збережить его в вашу домашню папку під назвою test.jpg.

В магістерській роботі для управління камерою буде використано Python та бібліотека `picamera`, що дає повний контроль над можливостями попереднього перегляду, отримання зображення та захват відео модулем камери. Саме за допомогою цих інструментів ми зможемо інтегрувати камеру в свій магістерський проект.

### **3.10 Створення та підключення схеми.**

Для мого проекту нам потрібна буде кнопка переключатель, макетка плата, перемикач «папа-папа» (**M2M**) та декілька перемичок «мама-папа» (**M2F**). Якщо немає макетної плати то можна підключити комунікатор, використавши замість цього кабель «мама-мама» (**F2F**), но його буде ваще натискати.

Для початку створемо схему Рисунок 3 – 20: додаємо кнопку на макетну плату, потім підключаємо шину заземлення до контакту заземлення на Raspberry Pi (відмічений **GND**) за допомогою перемички «папа-мама». Використовуємо

перемичку «папа-папа», щоб підключити одну ножку коммутатора до шини заземлення на мекетній платі, потім перемичку «папа-мама». щоб підключити другу ножку коммутатора до контакту 2 **GPIO** (відмічений **GP2**).

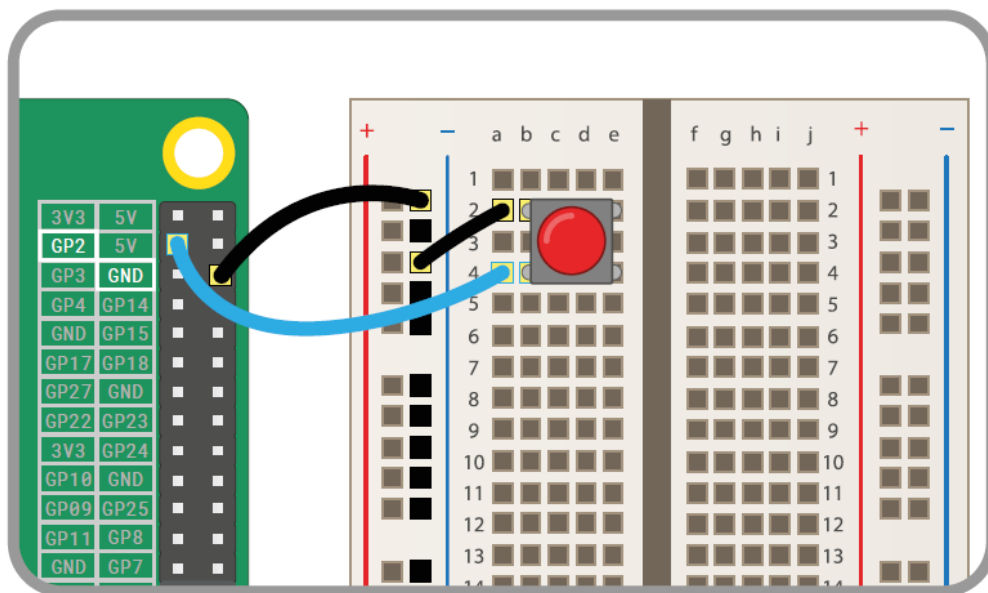


Рисунок 3.20 – Схема проекту.

Створення нового проекту в Thonny та збережіть його як **Stop Motion**. Рисунок 3 – 21. Почнемо з імпорту та налаштуванню бібліотек, Рисунок – 3.22, необхідних для використання камери та порту GPIO:

```
from picamera import PiCamera
from gpiozero import Button
camera = PiCamera()
button = Button(2)
```

Рисунок 3.21 – Налаштування бібліотеки.

Потім напишемо наступне:

```
camera.start_preview()
button.wait_for_press()
camera.capture('/home/pi/Desktop/image.jpg')
camera.stop_preview()
```

Рисунок 3.22 – Імпорт бібліотек.

Натиснемо «Виконати» (Run), і побачимо вікно попереднього переднього перегляду камери. Попередній перегляд буде залишатись на екрані до тих пір, поки не натиснемо кнопковий перемикач: як тільки його натиснемо, попередній перегляд закриється після того як наша програма збереже зображення на робочому столі. Найдемо зображення на робочому столі з іменем image.jpg та двічі натиснемо на нього, щоб підтвердити, що програма працює коректно.

Покадрова анімація пропонує створити багато нерухомих зображень, щоб створити ілюзію руху, коли вони зібрані всі разом та відтворюються по черзі.

### 3.11 Розширені налаштування камери

Якщо вам потрібен більший контроль над модулем камери Raspberry Pi, ви можете використовувати бібліотеку Python **picamera** для доступу до налаштувань. Ці параметри разом зі значеннями за замовчуванням докладно описані нижче, а вивчивши, ви зможете їх використовувати у ваших власних програмах.

```
camera.awb_mode = 'auto'
```

Ця настройка встановлює режим автоматичного балансу білого камери і може бути встановлений в будь-якій з наступних режимів: виключений, авто, сонячне світло, хмарно, тінь, вольфрам, флюоресцентний, лампа розжарювання, спалах або горизонт (відповідно: off, auto, sunlight, cloudy, shade, tungsten, fluorescent, incandescent, flash або horizon). Якщо ви виявите, що ваші фотографії і відео виглядають трохи синіми або жовтими, спробуйте застосувати інший режим.

```
camera.brightness = 50
```

Ця настройка встановлює яскравість зображення камери від самого темного при «0» до самого яскравого при «100».

```
camera.color_effects = None
```

Цей параметр змінює колірний ефект, який використовується камерою. Зазвичай цей параметр не слід змінювати, але якщо ви введете пару чисел,

то зможете змінити спосіб запису кольору камерою. Наприклад, спробуйте (128, 128), щоб створити чорно-біле зображення.

```
camera.contrast = 0
```

Ця настройка встановлює контрастність зображення. Чим більше число, тим більш драматичніше і різкіше виглядають речі; менше число зробить речі більш розмитими. Ви можете використовувати будь-яке число від «-100» для мінімального контрасту до «100» для максимального контрасту.

```
camera.crop = (0.0, 0.0, 1.0, 1.0)
```

Ця настройка дозволяє обрізати зображення, відрізаючи частини з боків і зверху, щоб захопити тільки ту частину зображення, яка вам потрібна. Числа представляють координату X, координату Y, ширину і висоту та за замовчуванням захоплює зображення. Спробуйте зменшити останні два числа «0,5» і «0,5» щоб побачити, який вплив на зображення має цей параметр.

```
camera.exposure_compensation = 0
```

Ця настройка встановлює компенсацію експозиції (exposure compensation) камери, дозволяючи вручну контролювати кількість світла, уловлюваного для кожного зображення. На відміну від зміни яскравості, це фактично керує самою камерою. Допустимі значення варіюються від «-25» для дуже темного зображення до «25» для дуже яскравого зображення.

```
camera.exposure_mode = 'auto'
```

Ця настройка встановлює режим експозиції (exposure mode), який модуль камери використовує щоб вирішити, як слід експонувати зображення. Можливі режими: вимк., авто, ніч, підсвічування, прожектор, спорт, сніг, пляж, дуже довга,

фіксована частота кадрів, антишейк і феєрверк (відповідно: off, auto, night, backlight, spotlight, sports, сніг, пляж, verylong, fixedfps, antishake і fireworks).

```
camera.framerate = 30
```

Ця настройка встановлює кількість зображень, захоплених для створення відео в секунду, або іншими словами — частоту кадрів (frame rate). Більш висока частота кадрів створює більш плавне відео, але займає більше місця для зберігання файлу. Більш висока частота кадрів вимагає використання більш низького дозволу, яке ви можете встановити за допомогою параметра camera.resolution.

```
camera.hflip = False
```

Ця настройка перевертає зображення камери по горизонтальній осі (осі X), якщо встановлено значення True.

```
camera.image_effect = 'none'
```

Ця настройка застосовує до відеопотоку один з ефектів зображення, який буде відображатися в попередньому перегляді, а також у збережених зображеннях і відео. Можливі ефекти: розмиття, мультиплікація, колірний баланс, колорпойнт, зміна кольорів, деінтерлейсинг1, деінтерлейсинг2, шумозаглушення, тиснення, плівка, грєп, штрихування, негатив, немає, олійна фарба, пастель, постеризація, насиченість, ескіз, соляризація, розмиття і акварель (відповідно: blur, cartoon, colorbalance, colorpoint, colorswap, deinterlace1, deinterlace2, denoise, emboss, film, gpen, hatch, negative, none, oilpaint, pastel, posterise, saturation, sketch, solarize, washedout і watercolor).

```
camera.ISO = 0
```

Цей параметр змінює налаштування ISO камери, що впливає на її чутливість до світла. За замовчуванням камера регулює ISO автоматично в залежності від доступного їй світу. Ви можете встановити ISO самостійно, використовуючи один з



наступних значень: 100, 200, 320, 400, 500, 640, 800. Чим вище ISO, тим краще камера буде працювати в умовах низької освітленості, але тим більш зернистим буде зображення або захоплене відео.

```
camera.meter_mode = 'average'
```

Ця установка впливає на те, як камера визначає кількість доступного світла при настройці експозиції. Значення за замовчуванням «average» (середнє) усереднює кількість світла, доступного по всьому зображенню. Інші можливі режими — «backlit» (з підсвічуванням), «matrix» (матричний) і «spot» (точковий).

```
camera.resolution = (1920, 1080)
```

Ця настройка встановлює дозвіл захопленого зображення або відео, представлене двома числами (для ширини та висоти відповідно). Більш низькі дозволу дозволяють займати менше місця для зберігання, а також використовувати більш високу частоту кадрів. Більш високий дозвіл означає кращу якість, але займає більше місця для зберігання.

```
camera.rotation = 0
```

Ця настройка управляє поворотом зображення від 0 градусів до 90, 180 і 270 градусів. Використовуйте її, якщо ви не можете розташувати камеру так, щоб стрічковий кабель модуля камери виходив у напрямку вниз.

```
camera.saturation = 0
```

Цей параметр контролює **насиченість зображення** або іншими словами — яскравість кольорів. Можливі значення «-100» до «100».

```
camera.sharpness = 0
```

Цей параметр контролює **різкість зображення**. Можливі значення «-100» до «100».

```
camera.shutter_speed = 0
```

Цей параметр контролює те, як швидко відкривається і закривається засувка (shutter) при зйомці зображень і відео. Ви можете встановити витримку (shutter speed) вручну в мікросекундах, при цьому більш довгі витримки краще працюють при слабкому освітленні, а більш короткі витримки — при більш яскравому. Зазвичай слід залишити значення за умовчанням — автоматичне («0»).

```
camera.vflip = False
```

Ця настройка, подібно до розглянутої вище hflip, перевертає зображення камери, але по горизонтальній осі (осі Y), якщо встановлено значення True.

```
camera.video_stabilization = False
```

Якщо встановлено значення True, включається стабілізація відео. Це необхідно тільки в тому випадку, якщо модуль камери рухається під час запису, наприклад, якщо він прикріплений до робота або переноситься на будь-якому обладнанні, щоб зменшити хиткість захопленого відео.

Ми розглянули основні функції настройки камери. Більш детальну інформацію про цих настройках, а також про додаткових настройках, які не були розглянуті тут, можна знайти на сайті бібліотеки picamera: [picamera.readthedocs.io](http://picamera.readthedocs.io).

На офіційному сайті Raspberry Pi в розділі проектів на Python для модуля камери ви можете спробувати свої сили в практичному застосуванні своїх знань в програмуванні мовою, а також використання модуля камери, просто дотримуючись докладним інструкцій, копіюючи написаний до вас код і перевіряючи заявлену в проектах функціональність. Ви можете перевірити на практиці такі проекти, як: зйомка в таймлапс, накладення масок на фотографії в реальному часі, створення об'єктів для Minecraft і багато іншого. Переглядаючи готові проекти і намагаючись повторити їх у себе на комп'ютері, ви дуже швидко підвищите свої знання та

навички роботи з Raspberry Pi, зовнішніми модулями і платами розширення. Існує також величезна кількість проектів, написаних на інших мовах програмування, наприклад, Scratch та інших. Це зроблено для вашої зручності і більшої гнучкості в застосуванні мікрокомп'ютерів Raspberry Pi для реалізації різноманітних задач в реальному світі.

### 3.12 Область застосування міні-ПК Raspberry Pi

- Використання в якості повноцінного ПК. Рисунок 3.23 – Повністю підключений міні-ПК. Можете спробувати підключити монітор, клавіатуру, мишку і отримати стаціонарний ПК на базі Windows 10. Однак, для цього підійдуть тільки характеристики Raspberry Pi B, а ще краще - 2B або 3B;
- Зібрати разом десятки міні-ПК Raspberry Pi і отримати багатоядерний ПК. Для цього з'єднаєте в один сервер потрібну кількість пристроїв, знайдіть для них відповідне місце і, бажано, охолодження. Багатоядерність такого сервера перевершить будь-сучасний процесор за сотні доларів;
- Встановіть на одноплатний комп'ютер Raspberry емулятор улюбленої консолі, скачайте пак образів для ігор, підключіть джойстик, монітор і насолоджуйтеся ігровою консоллю в міні-форматі;
- Підключіть міні-ПК до сенсорного екрану з діагоналлю від 8 до 15 дюйм, зробіть корпус з металу або дерева, встановіть операційну систему Android і отримаєте повноцінний планшет, створений своїми руками;
- Створіть власну систему освітлення для будинку або вулиці, налаштувавши Raspberry для чергування включення певних ламп, і здивуйте близьких;

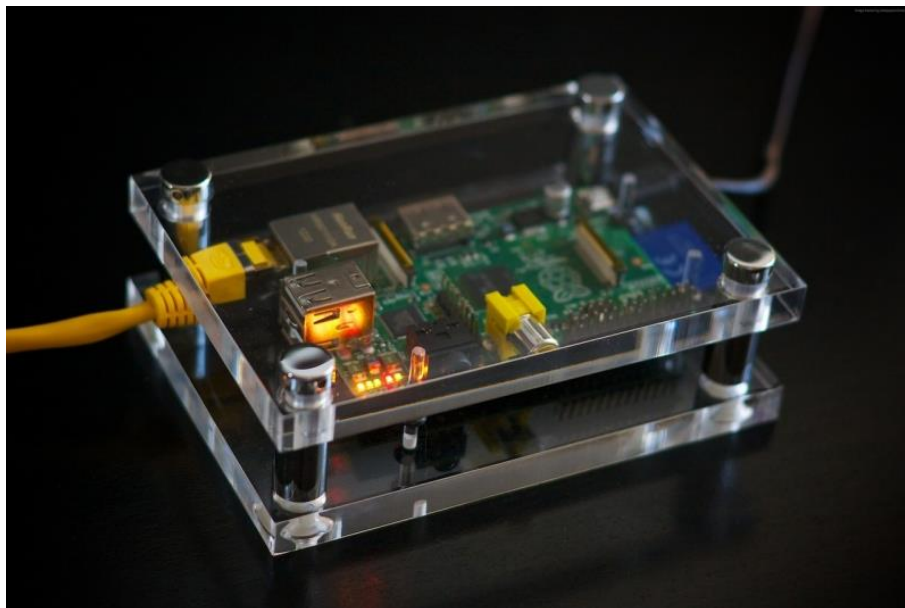


Рисунок 3.23 – Повністю підключений міні-ПК

- Встановіть на Raspberry Pi Skype, підключіть його до стаціонарного телефону, сховайте в корпусі і дзвоніть друзям з усього світу. Правда, для цього потрібно ще стабільне підключення до Wi-Fi і встановлена система Android або Windows;
- Створіть настінний органайзер, підключивши міні-ПК до невеликого екрану, налаштувавши операційну систему і вмонтувавши все це в стіну.

### **Висновок до третього розділу**

В третьому розділі описується алгоритм роботи, та способи керування системою детектування руху. Розглянуто вбудовані функції для виконання поставленого завдання, та наведено приклади роботи деяких функцій.

Представлено результати завантаження програмного коду для системи детектування руху на основі Raspberry Pi.

## ВИСНОВКИ

Результатом виконання магістерської роботи є система детектування руху на основі Raspberry Pi. Всі поставлені технічні завдання були вирішені, а саме:

- Було вивчено літературу по платформі Raspberry Pi і автоматичного детектування руху, що дає бачення шляхів вирішення проблеми автоматичного запису.
- Було створено прототип системи детектування руху на базі Raspberry Pi.
- Досліджено оптимальні параметри функціонування системи в невеликому приміщенні.
- Вирішино ряд проблем щодо використання системи детектуванні в умовах тривалої відсутності людини в приміщенні.
- Визначено напрями апаратного і програмного вдосконалення даної установки.
- Також було заплановано нові напрямки проектування та створення «Розумного будинку» та інших проектів на основі платформи Raspberry Pi.
- Система працює від джерела живлення з низькою напругою.

## Додаток А. Вихідний код програми

```
#імпортувати необхідні пакети
from gpiozero import Button, MotionSensor
from picamera import PiCamera
from time import sleep
from signal import pause

#створити об'єкти, які ссилаються на кнопки керування
#датчик руху та PiCamera
button = Button(2)
pir = MotionSensor(4)
camera = PiCamera()

#запустити камеру
camera.rotation = 180
camera.start_preview()

#Для назви зображень
i = 0

#зупинити роботу камери при натисненні кнопки
def stop_camera():
    camera.stop_preview()
    #вихід з програми
    exit()

#зробити фото при виявленні руху
def take_photo():
    global i
    i = i + 1
    camera.capture('/home/pi/Desktop/image_%s.jpg' % i)
    print('A photo has been taken')
    sleep(10)

#назначити функцію, яка запускається при натисненні кнопки
button.when_pressed = stop_camera
#назначити функцію, яка запускається при виявленні руху
pir.when_motion = take_photo

pause()
```

## ПЕРЕЛІК ПОСИЛАНЬ

1. Банци М. Raspberry для початківців чарівників / М. Банци. — М.: Рід Груп, 2012. — 128 с.
2. Корольов, Н.В. Мікроконтролери ATME1 на ядрі Cortex. Розширення сімейства. / М.В. Корольов // Електроніка: наука, технологія, бізнес. -2013. - №1
3. 5. Петін Ст. А. Проекти з використанням мікрокомп'ютера Raspberry Pi. — СПб.: БХВ-Петербург, 2014. — 400 с.: іл.
4. Блум Джеремі Вивчаємо Raspberry Pi: інструменти і методи технічного чарівництва: Пер. з англ. — СПб.: БХВ-Петербург, 2015. — 336 с: іл.
5. Улли Соммер Програмування мікрокомп'ютерних плат Raspberry Pi. — СПб.: БХВ-Петербург, 2012. — 238 с.: іл.
6. Торо Карвінен, Кіммо Карвінен, Віллі Валтакари Робимо сенсори. Проекти сенсорних пристроїв на базі Raspberry Pi. — Вільямс, 2015. — 445 с.
7. Amariei C. – Raspberry Pi Development Cookbook — Packt Publishing, 2015. — 246 p.
8. Julien Bayle - C Programming for Raspberry Pi — Packt Publishing, 2015. — 512p.
9. Керівництво по освоєнню Raspberry Pi — Creative Commons, 2012. — 36с.
10. Черничкин М.Ю. - Велика Енциклопедія Електрика — Эксмо, 2011. — 271с.
11. Evans Brian - Raspberry Pi блокнот програміста — Пер. с англ. Гололобов В.Н., 2007. – 40 с.
12. Р. Стюарт Болл - Аналогові інтерфейси мікроконтролерів —Додэка-XXI, 2007 — 360с.
13. Програмування Raspberry Pi. Режим доступу: <https://all-RaspberryPi.ru/programmirovanie-arduino/>
14. Датчики та модулі Raspberry Pi. Режим доступу
15. Блум Джеремі Вивчаємо Raspberry Pi: інструменти і методи технічного чарівництва: Пер. з англ. — СПб.: БХВ-Петербург, 2015. — 249 с: іл.



Кафедра комп'ютерної інженерії  
Система детектування руху на основі ESP8266-CAM та Raspberry Pi