

Tervezés Mikrovezérlőkkel 2023-2024

ESP32-CAM képfelvétel mozgásérzékelésre és Telegram chatbot-al való kommunikáció

Projekt

Hallgató: Vezetőtanár:

Molnár Orsolya-Izabella, Dr. Losonczi Lajos

Számítástechnika IV.

# TARTALOMJEGYZÉK

[**Bevezető**](#_ybqomksn158h) **3**

[**Felhasználói platform (ESP32-CAM) bemutatása**](#_z76vhz3kiz4m) **3**

[Általános tudnivalók](#_lmj8aha03xvs) 4

[Projekt specifikus tudnivalók](#_wr98euehlda5) 4

[Fejlesztői támogatás](#_ea7chlcet5on) 8

[Alkatrész lista](#_i00trqicfsue) 9

[Beavatkozó egység](#_fr8tk11i23g4) 9

[Feldolgozó egység](#_pnphrhvxdoxk) 10

[Érzékelő egység](#_53pby7brlaf1) 10

[Monitorizáló egység (Felhasználói felület)](#_x2xrhe1m3id5) 12

[**Megvalósítás és működés leírása**](#_sgpitxkyu75s) **12**

[Megvalósított rendszer leírása](#_a5m0o9zdr2h4) 14

[Arduino IDE beállítása ESP32-hez](#_3debsulqkali) 15

[Telegram bevonása](#_dverotk46k4q) 16

[**Lehetőségek a továbbfejlesztésre**](#_eluznmvd1o17) **21**

[**Program magyarázata**](#_x8let2o6dfz4) **22**

[**Demonstráció**](#_886edy36jn3g) **27**

[**Könyvészet**](#_e4klri2bote7) **31**

# 

# 

# 

## Bevezető

Ez a dokumentáció az 'ESP32-CAM képfelvétel mozgásérzékelésre és Telegram chatbot-tal való kommunikáció' projekt részletes bemutatására szolgál. A projekt lényegét az ESP32-CAM mikrovezérlő kiterjedt elemzése alkotja, melynek során az architektúrát nem csak megismerjük, hanem a gyakorlatban is alkalmazzuk. A célunk, hogy széleskörű betekintést nyújtsunk az ESP32 platformba, kiemelve annak alkalmazási lehetőségeit egy konkrét és életszerű példán keresztül.

A projekt középpontjában az ESP32-CAM mikrovezérlő áll, mely magában foglal egy WiFi modult és kamerát. A fő szempont az, hogy hogyan használhatjuk ki ezt a technológiát a mindennapi életben. A fő célok közé tartozik a mozgásérzékelés és képfelvétel, melyeket a beépített PIR érzékelő segítségével valósítunk meg.

A dokumentáció részletesen kitér majd az ESP32-CAM és PIR érzékelő hardver komponenseire, azok szükséges beállításaira, valamint a projekt telepítésére és konfigurálására az Arduino IDE használatával. Emellett ismertetni fogjuk a Telegram bot létrehozásának lépéseit, és hogyan integrálhatjuk azt az ESP32-CAM alkalmazásába.

A használat során bemutatom, hogyan küldhetünk parancsokat a Telegram botnak, és hogyan reagál az ESP32-CAM mozgásérzékelésekre, elküldve a megfelelő pillanat képeket a felhasználóknak. A dokumentáció részét képezik a lehetséges hibaelhárítási lépések is, amelyek segíthetnek a projekt zökkenőmentes működésének biztosításában.

Végül a dokumentáció kitér a jövőbeli fejlesztési lehetőségekre, például további érzékelők hozzáadására, a képek tárolásának és archiválásának lehetőségeire, valamint egy felhasználóbarát konfigurációs felület kialakítására.

## Felhasználói platform (ESP32-CAM) bemutatása

Az ESP32-CAM, mint rendkívül sokoldalú mikrovezérlő egység, egyre növekvő népszerűsége miatt kiemelkedő lehetőségeket kínál az elektronikai fejlesztők és a beágyazott rendszerek területén. Az eszköz ötvözi a hatékony mikrovezérlő teljesítményét egy beépített WiFi modullal és kamerával, így ideális választás olyan alkalmazásokhoz, ahol a vezeték nélküli kapcsolat és a képfelvétel együttesen szükséges.

Az ESP32-CAM széles körű funkcionalitása lehetővé teszi számos projekt megvalósítását, a különböző szenzorok integrálásától kezdve a kommunikációs platformokig. Emellett a könnyű programozhatóság és az Arduino IDE támogatásának köszönhetően a fejlesztők gyorsan és hatékonyan hozhatnak létre egyedi alkalmazásokat.

Ezen felül, az ESP32-CAM rendkívül költséghatékony megoldást nyújt, így ideális választás lehet a projektjeidhez anélkül, hogy a teljesítményt vagy a funkcionalitást áldoznád be. A dokumentáció további részeiben részletesen foglalkozunk a mikrovezérlő hardver jellemzőivel és alkalmazási területeivel, bemutatva, hogy miként lehet az ESP32-CAM-et hatékonyan kihasználni a különféle elektronikai és beágyazott rendszer fejlesztések során.

### Általános tudnivalók

Az ESP32 egy kétmagos mikrovezérlő, amelyet az Espressif Systems fejlesztett ki, és széles körben alkalmaznak az IoT (Internet of Things) alkalmazásokban. Az ESP32 fejlesztői kártya egy olyan platformot nyújt, amely lehetővé teszi a fejlesztők és mérnökök számára, hogy kihasználják az ESP32 teljesítményét és széles körű funkcióit. Itt találsz néhány kulcsfontosságú információt az ESP32 fejlesztői kártyáról:

1. Mikrovezérlő és Teljesítmény:

Az ESP32 két magos Xtensa LX6 mikrovezérlővel rendelkezik, amelyek akár 240 MHz-es órajelen is működhetnek. Ez lehetővé teszi a párhuzamos végrehajtást és a gyors reakciót.

1. Vezeték nélküli kapcsolódás:

Beépített WiFi modullal rendelkezik, támogatva a 802.11 b/g/n szabványokat, és két üzemmódban használható: AP (Access Point) és Station (állomás).

1. Bluetooth és BLE (Bluetooth Low Energy):

Az ESP32 támogatja a Bluetooth és BLE funkciókat, lehetővé téve a vezeték nélküli kommunikációt más eszközökkel.

1. Széles GPIO Tüskesorok:

Az ESP32 rendelkezik számos GPIO tüskesorral, amelyeket külső eszközök, érzékelők vagy kijelzők csatlakoztatásához lehet használni.

1. Programozhatóság:

Támogatja az Arduino IDE-t, PlatformIO-t és egyéb fejlesztői környezeteket, így könnyen programozható és konfigurálható.

1. Alacsony Energiafogyasztás:

Az ESP32 rendelkezik alacsony energiafogyasztású üzemmódokkal, amelyek lehetővé teszik az energiatakarékos alkalmazások kialakítását.

1. Szenzorok és Perifériák:

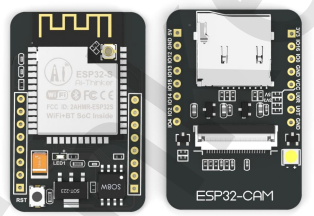
Az ESP32 számos beépített szenzort támogat, például Hall-szenzorokat, hőmérséklet-szenzort és egyebeket. Ezenkívül perifériákat is tartalmaz, mint például ADC, DAC, PWM, és I2C, SPI, UART interfészeket.

1. Támogatás és Közösség:

Az ESP32 rendelkezik egy aktív fejlesztői közösséggel és számos dokumentációval, ami segíti a fejlesztőket a projektek létrehozásában és problémák megoldásában.

### Projekt specifikus tudnivalók

A projekt megvalósításában egy ESP32-CAM típusú általános célú fejlesztői lapot használtam.



ESP32-CAM, forrás: <https://loboris.eu/ESP32/ESP32-CAM%20Product%20Specification.pdf>

Az ESP32-CAM egy rendkívül népszerű fejlesztői kártya, amelyet az Espressif Systems gyárt, és az ESP32 mikrovezérlő család egyik kiemelkedő tagja. Az Espressif Systems egy vezető mikrovezérlő és kommunikációs chip gyártó, kifejezetten specializálódva az IoT (Internet of Things) területén. Az ESP32-CAM alacsony költségű és kompakt méretű, így ideális választásnak számít a prototípusokhoz, DIY (Do It Yourself) projektekhez, valamint az IoT alkalmazásokhoz. A fejlesztők és a beágyazott rendszer tervezők széles körben alkalmazzák ezt a kártyát a vezeték nélküli kapcsolódás és a kamera funkciók kombinálásához a projektek számos területén.

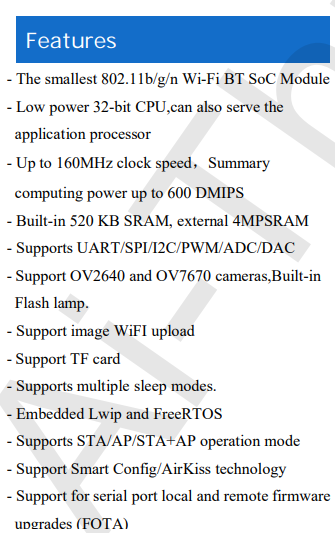
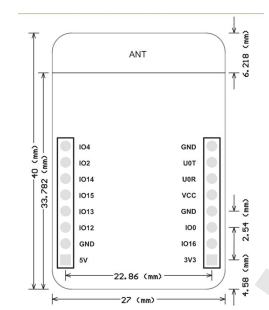
A kártya kétmagos LX6 CPU-t integrál, amelyeknek köszönhetően erős teljesítményt nyújt. A 7 szakaszos pipeline architektúra és az 80 MHz-től 240 MHz-ig terjedő fő frekvencia beállítási lehetőség optimális teljesítmény és energiafogyasztás egyensúlyát biztosítja.

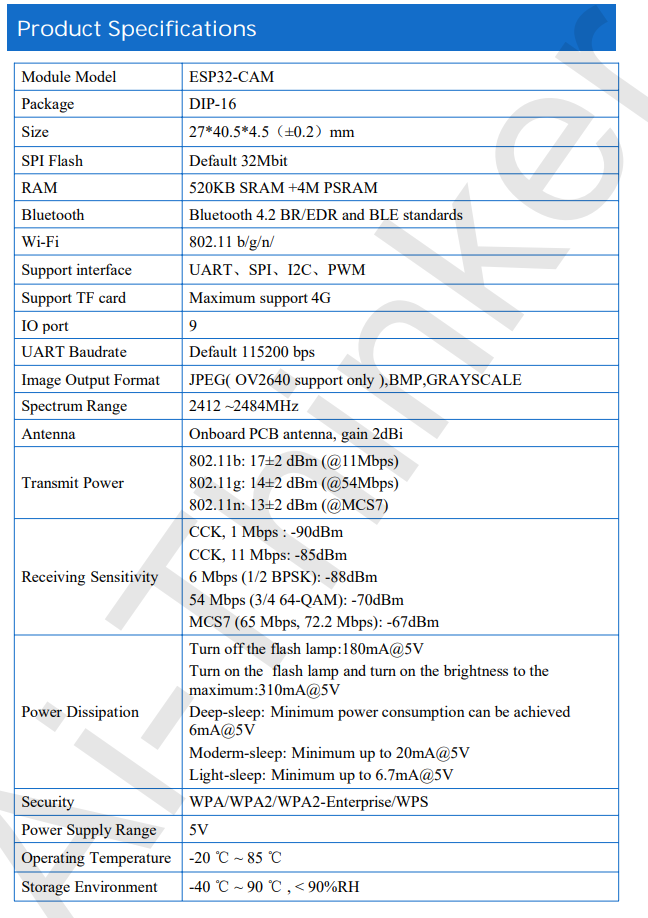
Az ESP32-CAM teljes mértékben kompatibilis a WiFi 802.11b/g/n/e/i és a Bluetooth 4.2 szabványokkal. Ezzel a vezeték nélküli kapcsolódás széles körű alkalmazásokban lehetséges. A kártya használható önálló hálózatvezérlőként vagy más vezérlő MCU-hoz csatlakoztatva, ezzel hálózati képességeket adva meglévő eszközöknek.

Az ESP32-CAM beépített mozgásérzékelővel és kamerával rendelkezik, lehetővé téve a képfelvételeket vagy videókészítést mozgás esetén. On-chip szenzorokkal, Hall-szenzorral és hőmérséklet-szenzorral további érzékelési lehetőségeket kínál.

Ez a kártya kiválóan alkalmazható otthoni okoseszközök, ipari vezeték nélküli vezérlés, vezeték nélküli monitorozás, QR-kódos vezeték nélküli azonosítás és más IoT alkalmazásokhoz. Az ESP32-CAM alacsony költségvetésű és kompakt méretű, így ideális eszköz prototípusokhoz és saját készítésű projektekhez is. A beépített kamera és a vezeték nélküli kapcsolódási lehetőségek révén széles körű alkalmazási területeket fed le.

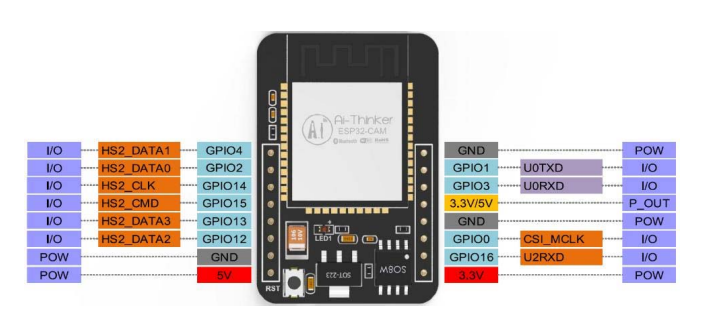
Egy listában összefoglalom röviden a jellemzőit:

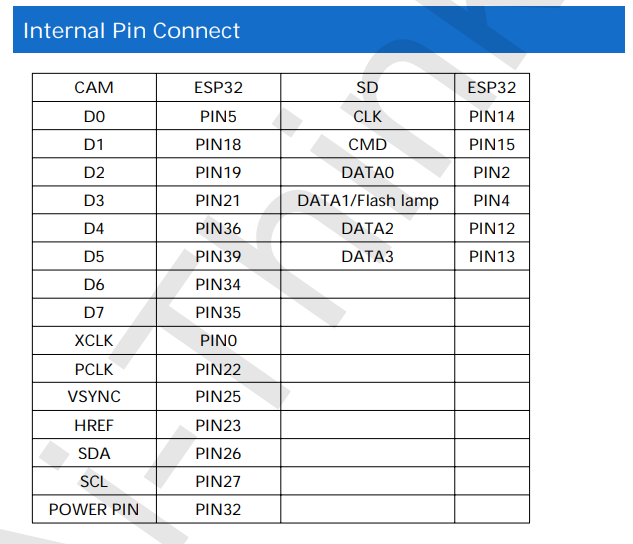


ESP32-CAM méretei és jellemzői, forrás: <https://loboris.eu/ESP32/ESP32-CAM%20Product%20Specification.pdf>

Az ESP32-CAM lábkiosztása a következő:



Forrás: <https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf>



Forrás: <https://loboris.eu/ESP32/ESP32-CAM%20Product%20Specification.pdf>

### Fejlesztői támogatás

A fejlesztés az Arduino IDE platformon történt.

Az Arduino IDE (Integrated Development Environment) számos előnnyel rendelkezik, különösen azok számára, akik kezdők vagy hobbifejlesztők az elektronikai és beágyazott rendszerek területén. Íme néhány ok, amiért az Arduino IDE-t jó választásnak tekinthetjük:

1. Az Arduino IDE rendkívül felhasználóbarát és könnyen kezelhető interfészt kínál. A kezdők számára egyszerűen érthető, ami segíti az alapvető kódolási és fejlesztési folyamatok megértését.
2. Az Arduino IDE gyorsan telepíthető és beállítható. Az új hardverek és fejlesztői platformok hozzáadása is egyszerű, például az ESP32 esetében.
3. Az Arduino IDE nyílt forráskódú, és aktív fejlesztői közösséggel rendelkezik. Ezáltal sokféle hardvert és kiegészítőt támogat, és folyamatosan frissülnek a rendelkezésre álló könyvtárak és példák.
4. Az Arduino IDE lehetővé teszi a könnyű firmware feltöltést a támogatott fejlesztői kártyákra. A "Upload" gomb megnyomásával a kód gyorsan átkerül a fejlesztői kártyára.
5. Az IDE beépített Serial Monitorral rendelkezik, amely lehetővé teszi a változók, hibajelentések és más információk közvetlen megtekintését a mikrovezérlőről érkező soros porton keresztül.
6. Az Arduino IDE számos előre elkészített könyvtárral rendelkezik, amelyek segítik a fejlesztőket a különböző szenzorok, kijelzők és más eszközök használatában.
7. Az Arduino IDE platformfüggetlen, tehát szinte bármilyen operációs rendszeren használható (Windows, macOS, Linux).
8. Az Arduino közösség aktív és segítőkész, így könnyen lehet segítséget találni vagy megosztani tapasztalatokat más fejlesztőkkel. Emellett a dokumentáció széles körben elérhető és jól kidolgozott.

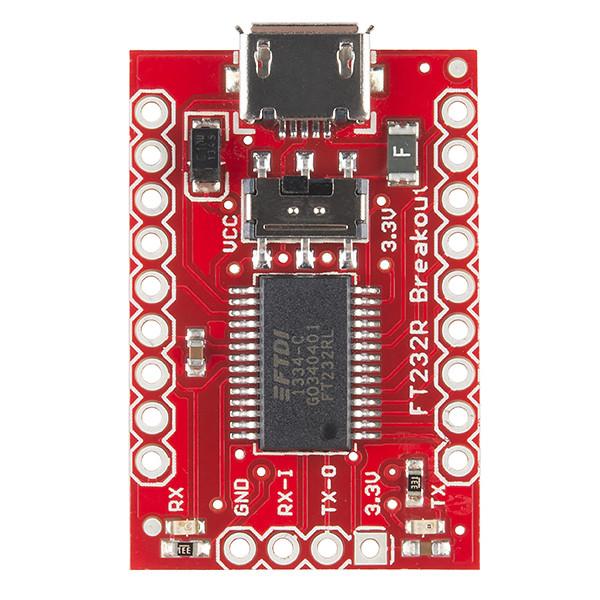
### Alkatrész lista

A rendszer több komponensből épül fel. Funkcionalitás szempontjából 4 csoport szerint osztályozhatjuk:

* Beavatkozó egység
* Feldolgozó egység
* Érzékelő egység
* Monitorizáló egység

#### Beavatkozó egység

Mivel az ESP32-CAM fejlesztői kártyának nincs dedikált programozó tüskesora, ezért a kód feltöltéséhez és annak frissítéséhez szükség volt egy különálló programozóra. Ez lehet pl. FTDI, Arduino UNO. Ezen projekten belül Arduino UNO-t használtam. Kezdetben FTDI programmerel szerettem volna megoldani, viszont több kód frissítés után, már nem működött.

Forrás: Google.com

#### Feldolgozó egység

A rendszer központi eleme a feldolgozó egység, a már korábban említett és bemutatott ESP32-CAM típusú mikrovezérlő. Azért esett erre a típusú egységre a választásom, mivel olcsó és a célnak megfelelő számítási kapacitással és kellő számú interface-szel (GPIO kimenettel) rendelkezik.

#### Érzékelő egység

A PIR (Passive Infrared) mozgásérzékelő olyan eszköz, amely képes érzékelni a környezetében lévő hőmérsékleti változásokat és mozgást. Ezek az érzékelők hasznosak különböző alkalmazásokban, beleértve a biztonsági rendszereket, világításvezérlést és egyéb automatizálási projekteket. Íme néhány kulcsfontosságú jellemző és információ a PIR mozgásérzékelőkről:

* A PIR érzékelő passzív eszköz, ami azt jelenti, hogy maga nem küld ki jelet, hanem csak észleli a környezeti infravörös sugárzást. Amikor egy objektum vagy személy áthalad a látómezőn, és a hőmérséklet változik, a PIR érzékelő kimeneti jelet generál.
* Az érzékelési távolság a PIR érzékelő hatótávolságát jelzi. Ez a távolság általában néhány métertől akár 10-15 méterig terjedhet attól függően, milyen típusú PIR érzékelőről van szó.
* A PIR érzékelők rendelkeznek egy meghatározott látószöggel, amely azt mutatja meg, hogy mekkora területet képesek érzékelni. A látószög mérete változhat a különböző típusú érzékelők esetében.
* A PIR érzékelők általában rendelkeznek érzékenységi beállítással és időzítő funkcióval. Az érzékenység beállítása meghatározza, milyen érzékenyek a környezeti hőmérsékleti változásokra. Az időzítő beállítása pedig azt szabályozza, mennyi ideig maradjon aktív az érzékelő a mozgás észlelése után.
* A PIR érzékelők széles körű felhasználási területekkel rendelkeznek, beleértve a biztonsági rendszereket, világításvezérlést, energiatakarékos alkalmazásokat és automatizálási projekteket. Például, ha a PIR észleli a mozgást egy szobában, aktiválhatja a világítást vagy más eszközöket.
* Vannak olyan PIR érzékelők, amelyek csak mozgást észlelnek, és olyanok, amelyek képesek megkülönböztetni az objektumok hőmérsékletét. Az utóbbi típusokat néha "két zónás" PIR érzékelőknek is nevezik.

Választásom az alábbi Mini AM312 PIR szenzorra esett:



Forrás: [www.emag.ro](http://www.emag.ro)

Az AM312 rendkívül kicsi és könnyű, ami lehetővé teszi a könnyű integrációt a különböző projektekbe anélkül, hogy túlzott teret foglalna el. A szenzor alacsony áramfogyasztással rendelkezik, ami fontos szempont lehet olyan projektekben, ahol az energiahatékonyság kiemelt szerepet játszik. Annak ellenére, hogy kis méretű, az AM312 rendelkezik egy elfogadható érzékelési távolsággal és látószöggel, ami lehetővé teszi a hatékony mozgásérzékelést egy adott környezetben. Az AM312 általában kedvező áron elérhető, ami lehetővé teszi a költséghatékony megoldások kialakítását projektek számára. Az AM312 könnyen integrálható a különböző fejlesztői környezetekkel és mikrovezérlőkkel, például az Arduino platformmal. A szenzor széles körű felhasználási területeken alkalmazható, beleértve az otthoni automatizálást, biztonsági rendszereket, világításvezérlést és egyéb projekteket.

Technikai jellemzők:

* Működési feszültség: DC 2,7V - 12V
* Statikus energiafogyasztás: <0,1mA
* Késleltetési idő: 2 másodperc
* Zárolási idő: 2 másodperc
* Érzékelési tartomány: 3m - 5m
* Működési hőmérséklet: -20℃ ~ +60℃
* Méretek: 12mm x 25 mm.

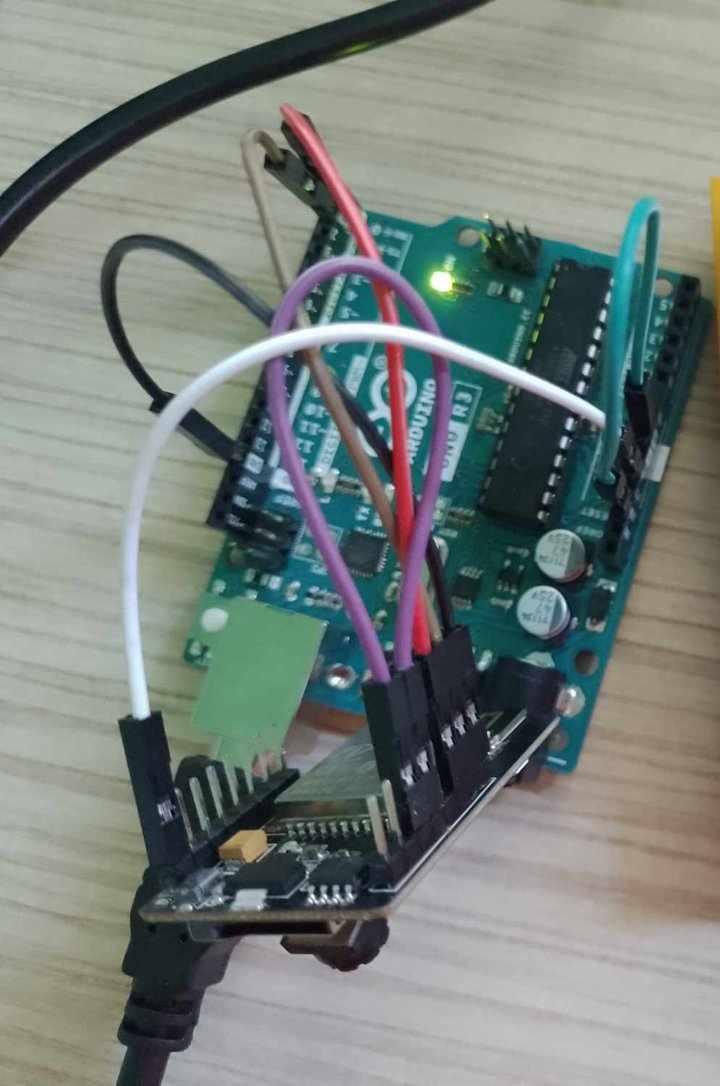
#### Monitorizáló egység (Felhasználói felület)

Egy Telegram bot-ot készítettem az ESP32-CAM-nek, amely engedélyezni tudja, hogy kontrolláljuk az eseményeket azaz, ha mozgást érzékel a szenzor, akkor az ESP készít egy képet, ami majd az applikációban megjelenik. Kezdetben egy értesítést kapunk. Mindezek mellett, Telegram-ból tudunk parancsokat kiküldeni a botnak, amivel tudjuk irányítani, hogy mi is történjen. Ezt a részt a későbbiekben fogom részletezni.

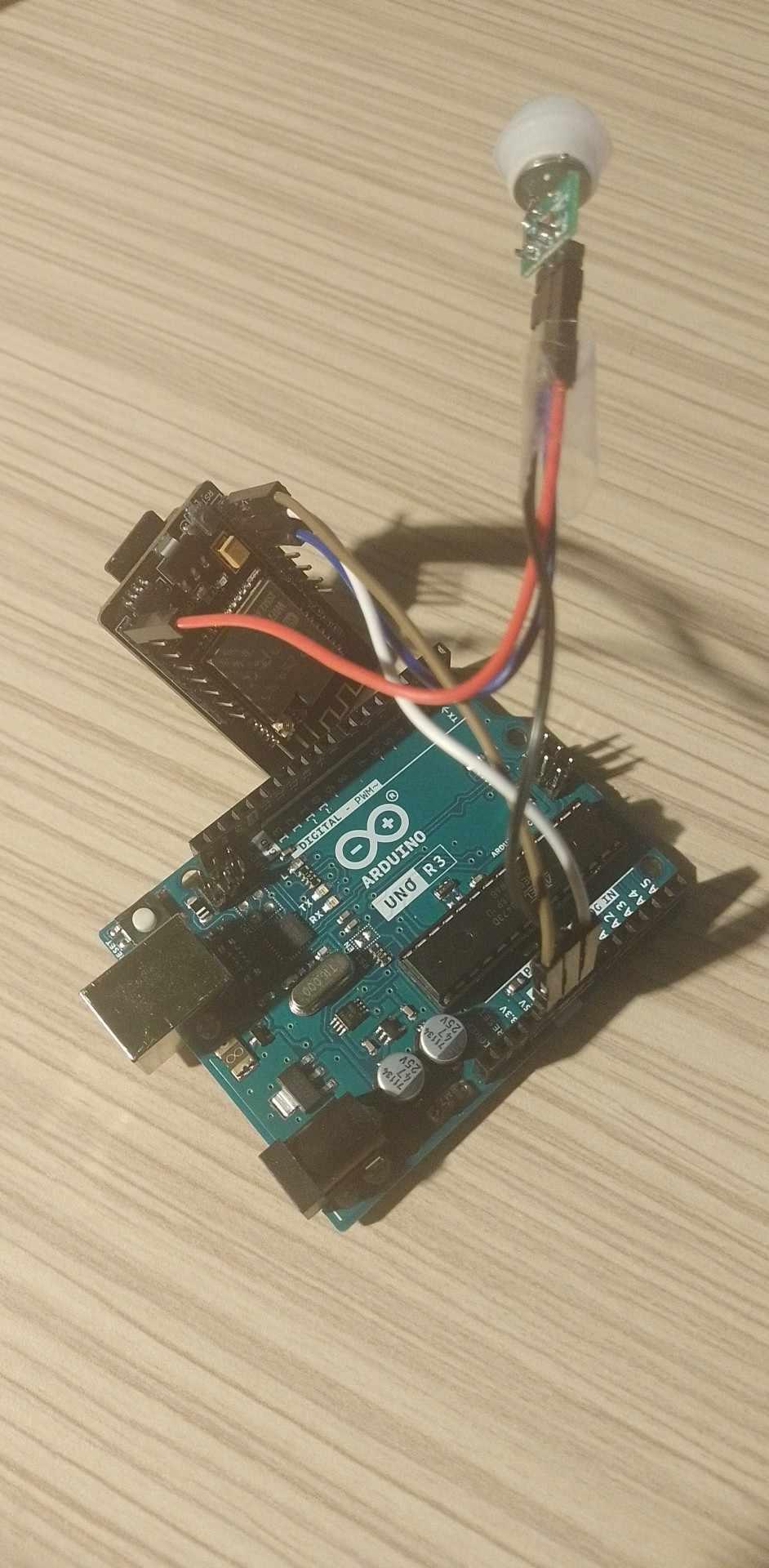
Ezáltal a projekt célja, hogy egy intelligens és interaktív rendszert hozzon létre, amely lehetővé teszi a felhasználók számára, hogy távolról ellenőrizzék és irányítsák az ESP32-CAM-et, és azonnal értesüljenek a mozgásérzékelés által rögzített eseményekről a Telegram alkalmazáson keresztül.

## Megvalósítás és működés leírása

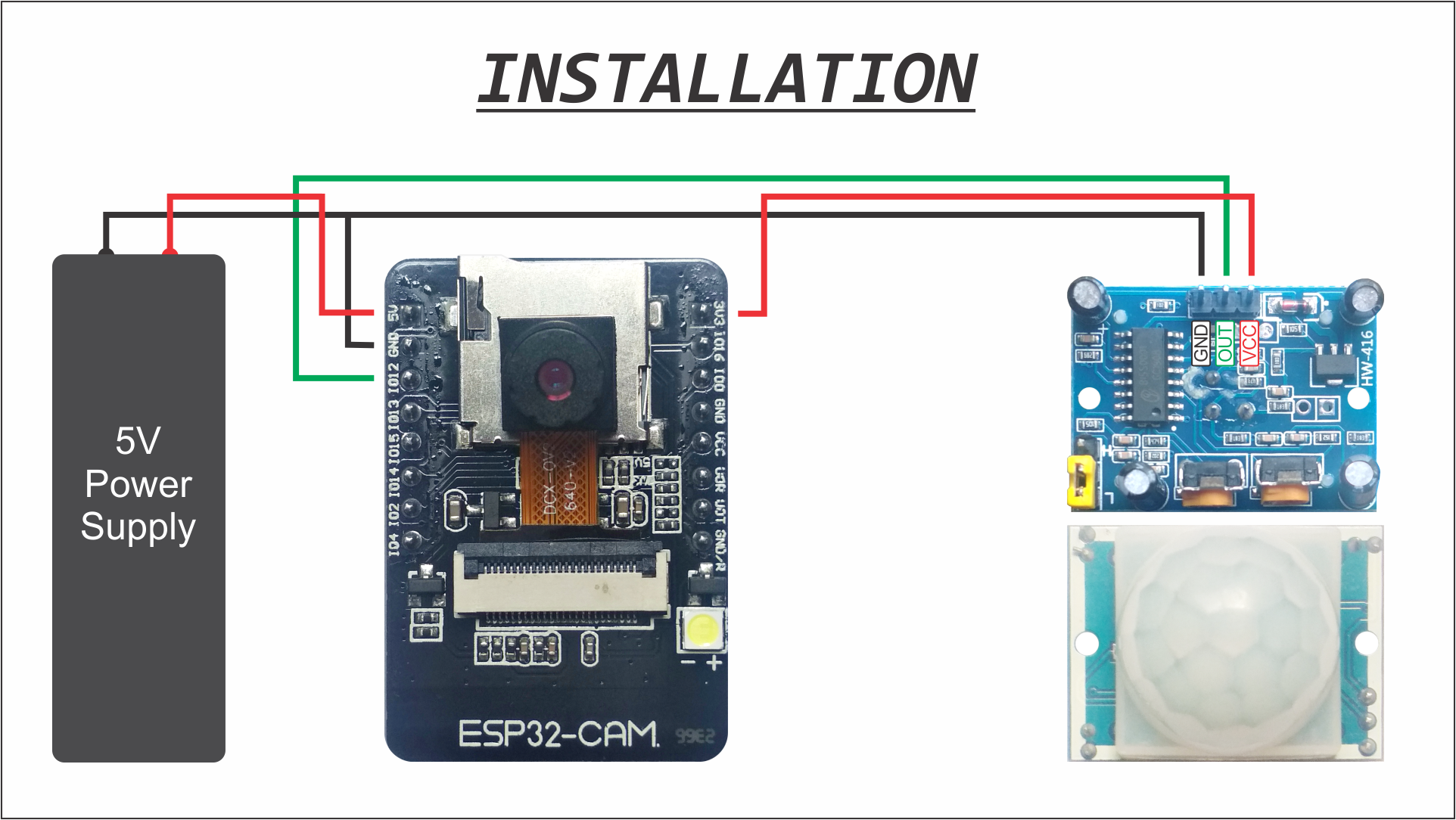
Az alábbi képen látható az az összeköttetés, amellyel felprogramoztam az ESP32-CAM-et.



Az alábbiakban pedig bemutatom a rendszer áramkörét:



Rendszer áramköre



Kamera összeköttetése a PIR szenzorral

### Megvalósított rendszer leírása

A tervezett rendszer az elképzelés és a megvalósítás között nem mutat sok különbséget. A megvalósítás során több problémába ütköztem, de többségében megoldódott és hasznos tanulságokkal gazdagodtam.

A végső rendszer működése a következők szerint alakult. A tervezetthez képest közel minden funkcionalitás megvalósításra került. Az apró módosítások csupán azért történtek, mert a fejlesztés során rájöttem, hogy jobb megoldások is léteznek az egyes feladatokra, pl. Az FTDI lecserélése Arduino UNO-ra, de még sok más javítás a forráskódra nézve.

Ez a projekt egy ESP32-CAM alapú rendszert valósít meg, amely képeket készít és továbbítja azokat a Telegram alkalmazásba. A rendszert továbbfejlesztettük egy PIR mozgásérzékelővel, amely észleli a környezetében történő mozgásokat, és kiváltja a képfelvételt, amit aztán az ESP32-CAM továbbít a Telegram botnak. A következőkben részletesen bemutatom a projekt működését.

Az ESP32-CAM és a PIR szenzor inicializációja és bekötése után a WiFi hozzáférést konfiguráljuk az ESP32-CAM kódjában, és beállítjuk a Telegram bot tokenjét. Ezzel a Telegram és az ESP32-CAM közötti kommunikáció létrejön.

A Telegram botot a BotFather segítségével hozzuk létre, és a kapott bot token segítségével a rendszer értesítéseket küldhet a Telegram alkalmazásba. A kamerakép elküldésekor az ESP32-CAM választható módon képet készít, majd továbbítja azt a Telegram botnak.

A Telegram alkalmazásban küldött parancsokkal lehetőség nyílik az ESP32-CAM vezérlésére. Például, a /capture parancs segítségével a rendszer képet készít, és elküldi azt a Telegramnak, ahol a felhasználó megtekintheti.

A PIR szenzor észleli a környezetében történő mozgást, és ennek hatására a rendszer azonnal értesítést küld a Telegram alkalmazásba. Az értesítés tartalmazza a mozgás tényét és a készült képet, lehetővé téve a felhasználók számára a gyors reakciót és a vizsgálatot.

#### Arduino IDE beállítása ESP32-hez

1. Telepítés: Először telepíteni kell az Arduino IDE-t a hivatalos webhelyről (<https://www.arduino.cc/en/software>).
2. Board Manager hozzáadása:

* A "File" menüből válaszd ki a "Preferences" lehetőséget.
* Másold be a következő URL-ket a "Additional Boards Manager URLs" mezőbe: <https://dl.espressif.com/dl/package_esp32_index.json>, http://arduino.esp8266.com/stable/package\_esp8266com\_index.json
* Kattints az "OK" gombra.

1. ESP32 Board Telepítése:

* A "Tools" menüből válaszd ki a "Board" opciót, majd a "Boards Manager" opciót.
* Keress rá az "esp32" szóra, és telepítsd fel az Espressif ESP32 platformot.
* Telepitsd az Universal Arduino Telegram Bot Lib mappát : https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot
* Telepítsd: ArduinoJson

1. Board Kiválasztása:

* A "Tools" menüből válaszd ki a "Board" opciót, majd válaszd ki az "ESP32 Wrover Module" lehetőséget.

1. Port Kiválasztása:

* Válaszd ki a megfelelő portot az "Tools" menü "Port" opciójából.

1. Sketch (Kód) Írása:

* Írj Arduino kódot a fő Sketch (program) fájlba. Az ESP32 specifikus funkciókat a hivatalos ESP32 Arduino könyvtárával éred el.

1. Feltöltés (Upload):
   * Kattints a "Upload" ikonra a kódot feltölteni. Ez fordítja a kódot és telepíti azt az ESP32-re.

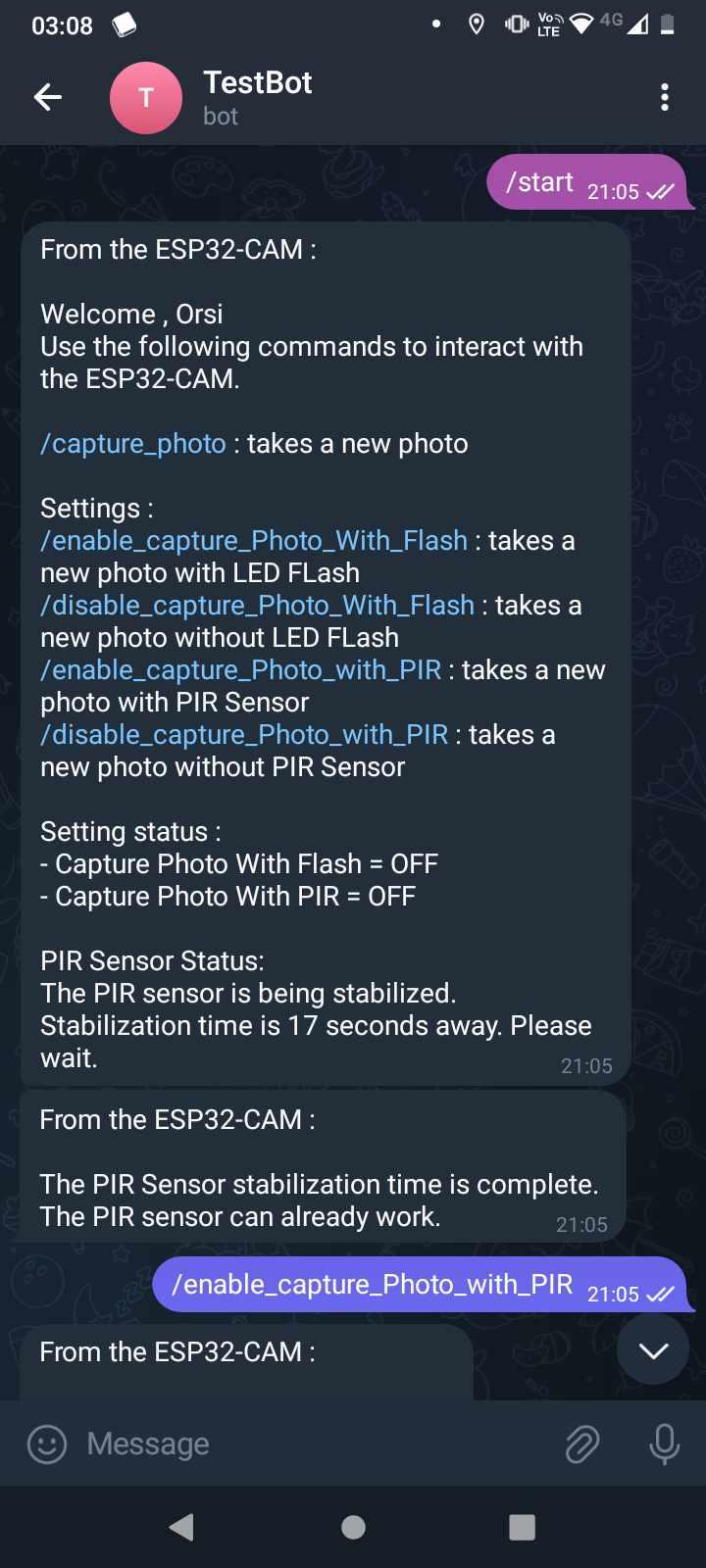
8. Nyisd Meg a Serial Monitort:

* A "Tools" menüből válaszd ki a "Serial Monitor" opciót.
* Itt ellenőrizheted a kimeneti üzeneteket és a hibajelentéseket.

### Telegram bevonása

Az ESP32-CAM pajzs ellenőrzéséhez létrehozunk egy Telegram botot, így bárhonnan monitorozhatod az ESP32-CAM-ot (feltéve, hogy internet-hozzáférésed van a smartphone-odonon). A következő parancsokat használhatod a botoddal való interakcióhoz:

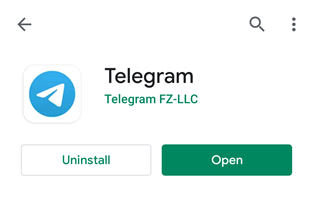
* /start: elküld egy üdvözlő üzenetet a pajzs vezérléséhez szükséges érvényes parancsokkal;
* /enable\_capture\_Photo\_With\_Flash : takes a new photo with LED FLash;
* /disable\_capture\_Photo\_With\_Flash : takes a new photo without LED FLash;
* /enable\_capture\_Photo\_with\_PIR : takes a new photo with PIR Sensor;
* /disable\_capture\_Photo\_with\_PIR : takes a new photo without PIR Sensor;



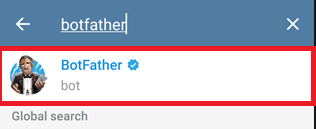
Ezen kívül, minden alkalommal értesítést kapsz egy fényképpel, amikor mozgást észlel a rendszer. Végül, csak te (vagy bármely más engedélyezett felhasználó, akit kiválasztasz) tudod vezérelni az ESP32-CAM-ot a Telegram segítségével.

Kövesd az alábbi lépéseket a Telegram bot létrehozásához.

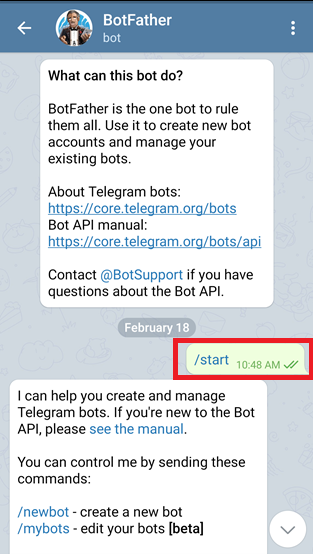
1. Lépj a Google Play vagy az App Store alkalmazásboltjába, töltsd le és telepítsd fel a Telegram alkalmazást.



1. Nyisd meg a Telegram alkalmazást, és kövesd az alábbi lépéseket, hogy létrehozz egy Telegram botot. Első lépésként keress rá a "botfather"-ra, majd kattints a BotFather nevére, ahogy az alábbiakban láthatod. Vagy nyisd meg ezt a linket a smartphone-odonon: [t.me/botfather](https://t.me/botfather).



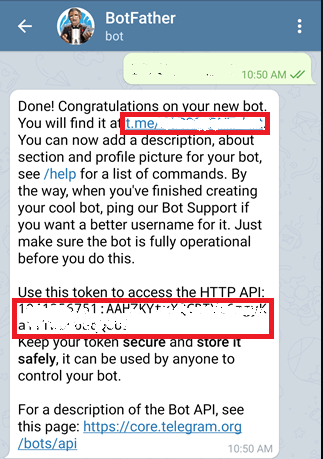
1. A következő ablaknak kell megjelennie, és fel fog kérni, hogy kattints a "start" gombra.



1. Írd be a /newbot parancsot, majd kövesd az utasításokat a bot létrehozásához. Adj neki egy nevet és egy felhasználónevet.



1. Ha a bot sikeresen létrejött, kapni fogsz egy üzenetet egy hivatkozással a bot eléréséhez, valamint a bot tokenjével. Mentsd el a bot tokent, mert szükséged lesz rá ahhoz, hogy az ESP32 interakcióba léphessen a botoddal.

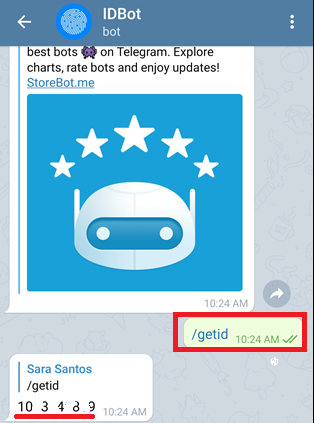


Bárki, aki ismeri a bot felhasználónevét, interakcióba léphet vele. Annak érdekében, hogy biztosítsuk, hogy csak azokat az üzeneteket vegyük figyelembe, amelyek az mi Telegram fiókunkból (vagy bármely más engedélyezett felhasználóból) származnak, szerezd be a Telegram felhasználói azonosítódat. Ezután, amikor a Telegram botod üzenetet kap, az ESP ellenőrizheti, hogy a küldő azonosítója megfelel-e a Te felhasználói azonosítódnak, és kezelheti az üzenetet vagy figyelmen kívül hagyhatja azt.

A Telegram fiókodban keress rá az "IDBot"-ra, vagy nyisd meg ezt a linket a smartphone-odonon: [t.me/myidbot](https://t.me/myidbot).



Kezdd el a beszélgetést ezzel a bottal, majd írd be a /getid parancsot. Válaszként megkapod a felhasználói azonosítódat. Mentsd el ezt a felhasználói azonosítót, mert később szükséged lesz rá ebben a bemutatóban.



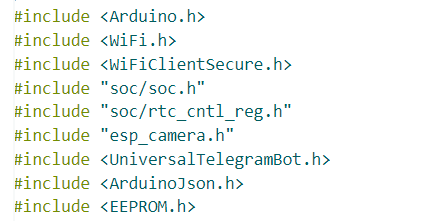
## Lehetőségek a továbbfejlesztésre

* Implementálhatunk további távoli vezérlési lehetőségeket a Telegram botban, például a kamera elforgatása vagy specifikus beállítások módosítása.
* Kiterjeszthetjük a rendszert úgy, hogy automatizált riasztásokat küldjön például e-mailen vagy más kommunikációs csatornákon keresztül, ha a PIR mozgásérzékelő jelentős eseményt észlel.
* További funkcionalitás lehet a személyazonosítás beépítése, amely lehetővé teszi a rendszer számára, hogy megkülönböztesse az ismert és ismeretlen személyeket a képek alapján.
* Integreálhatjuk a projektet felhőszolgáltatásokkal, például Google Drive-dal vagy Dropbox-szal, hogy a felvételeket tárolhassuk és könnyen hozzáférhessünk hozzájuk a felhőből.
* További érzékelők, például hőmérsékleti, páratartalmi vagy zajérzékelők hozzáadása a rendszer funkcionalitásának bővítéséhez.
* Készíthetünk mobilalkalmazást, amely lehetővé teszi a felhasználók számára a rendszer könnyebb és gyorsabb kezelését, valamint értesítéseket a mozgásokról.
* Kifejleszthetünk egy energiagazdálkodási mechanizmust, például alvó üzemmódot a felesleges energiafogyasztás minimalizálása érdekében.

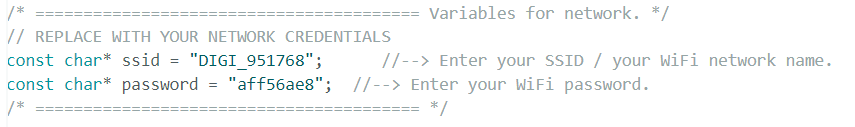
## Program magyarázata

Ez a szakasz magyarázza el, hogyan működik a kód.

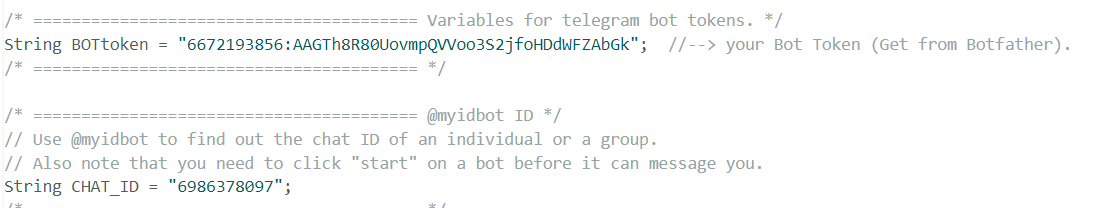
Könyvtárak hozzáadása:



WiFi-hez való kapcsolás (hálózati hitelek)



ChatBot-hoz való kapcsolódás:



A sendPhoto bool változó jelzi, hogy eljött-e az ideje egy új fénykép küldésének a telegram fiókjába. Alapértelmezés szerint hamisra van beállítva.



Hozzon létre egy új WiFi klienst a WiFiClientSecure segítségével.



Hozzon létre egy botot a korábban meghatározott tokennal és klienssel.



Hozzon létre egy változót a flash LED tű jelzésére (FLASH\_LED\_PIN). Az ESP32-CAM Wrover Module-ban a vaku a GPIO 4-hez csatlakozik. Alapértelmezés szerint állítsa be LOW-ra.

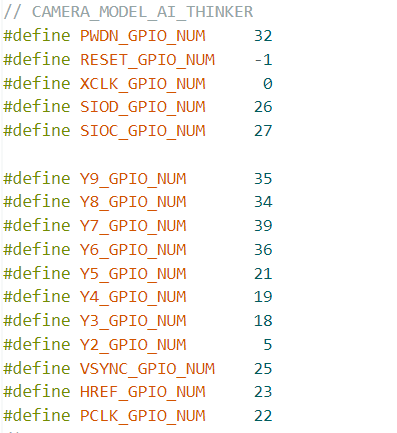
#define FLASH\_LED\_PIN 4

bool flashState = LOW;

A botRequestDelay és lastTimeBotRan változókat arra használják, hogy ellenőrizzék az új Telegram üzeneteket minden x másodpercben. Ebben az esetben a kód minden másodpercben (1000 milliszekundum) ellenőrzi az új üzeneteket. Ezt a késleltetési időt megváltoztathatja a botRequestDelay változóban.



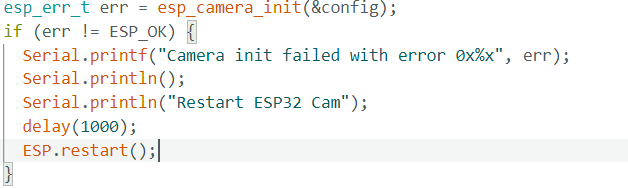
Kamera GPIO beállítások:



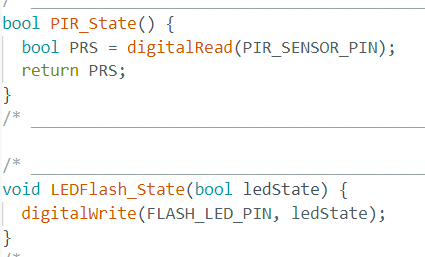
Kamera konfigurációk:



Kamera inicializálás:



LED és PIR szenzor vezérlése:



A bot üzeneteket a void handleNewMessage(int numNewMessages) függvény kezeli, míg a fényképek készitése és küldése a Telegram botnak a String sendPhotoToTelegram() függvény végzi.

*void handleNewMessages(int numNewMessages) {*

*Serial.print("Handle New Messages: ");*

*Serial.println(numNewMessages);*

Ellenőrzi az elérhető üzeneteket:

*for (int i = 0; i < numNewMessages; i++)*

Megszerezze a beszélgetési azonosítót egy adott üzenetért, és tárolja azt a chat\_id változóban. A chat ID azonosítja, ki küldte az üzenetet.

*String chat\_id = String(bot.messages[i].chat\_id);*

Ha a chat\_id eltér a saját chat ID-jétől (CHAT\_ID), az azt jelenti, hogy valaki más (nem te) küldött üzenetet a botodnak. Ebben az esetben figyelmen kívül hagyja az üzenetet, és várja meg a következő üzenetet.

*if (chat\_id != CHAT\_ID){*

*bot.sendMessage(chat\_id, "Unauthorized user", "");*

*continue;*

*}*

Ellenkező esetben az üzenet egy érvényes felhasználótól érkezett, így elmentjük a text változóban, majd ellenőrizzük annak tartalmát.

*String text = bot.messages[i].text;*

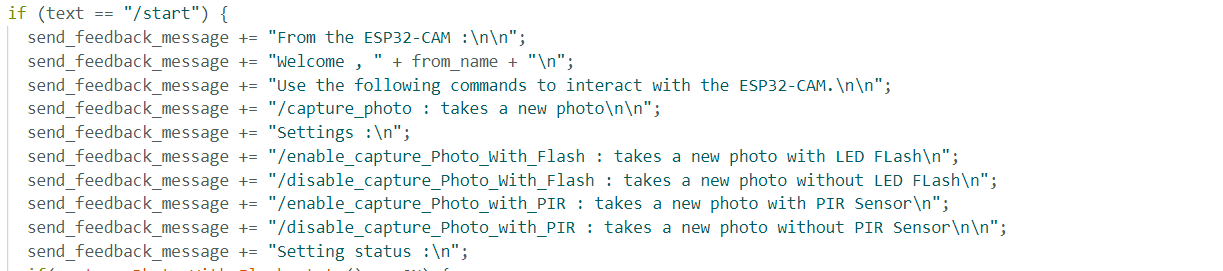
*Serial.println(text);*

A from\_name változó elmenti az üzenet küldőjének a nevét.

*String from\_name = bot.messages[i].from\_name;*

Ha a /start üzenetet kapja, akkor elküldi a érvényes parancsokat az ESP vezérléséhez. Ez hasznos, ha elfelejtetted, mi a parancs a táblád vezérléséhez.

Ezen kívül, a kód el van látva részletesen hozzászólásokkal, hogy minél érthetőbb legyen mindenki számára, hogy mi hogyan műkődik.



Az üzenet küldése a bótnak nagyon egyszerű. Csak használnod kell a sendMessage() metódust a bot objektumon, és megadni a címzett beszélgetési azonosítóját, az üzenetet és a parsolási módot (opcionális) argumentumként.



Egyéb funkciók: További segédfüggvények találhatók a kamera beállításainak mentésére és olvasására EEPROM-ból, valamint különböző beállítások kiírására a soros monitorra.

*void enable\_capture\_Photo\_With\_Flash(bool state);*

*bool capture\_Photo\_With\_Flash\_state();*

*void enable\_capture\_Photo\_with\_PIR(bool state);*

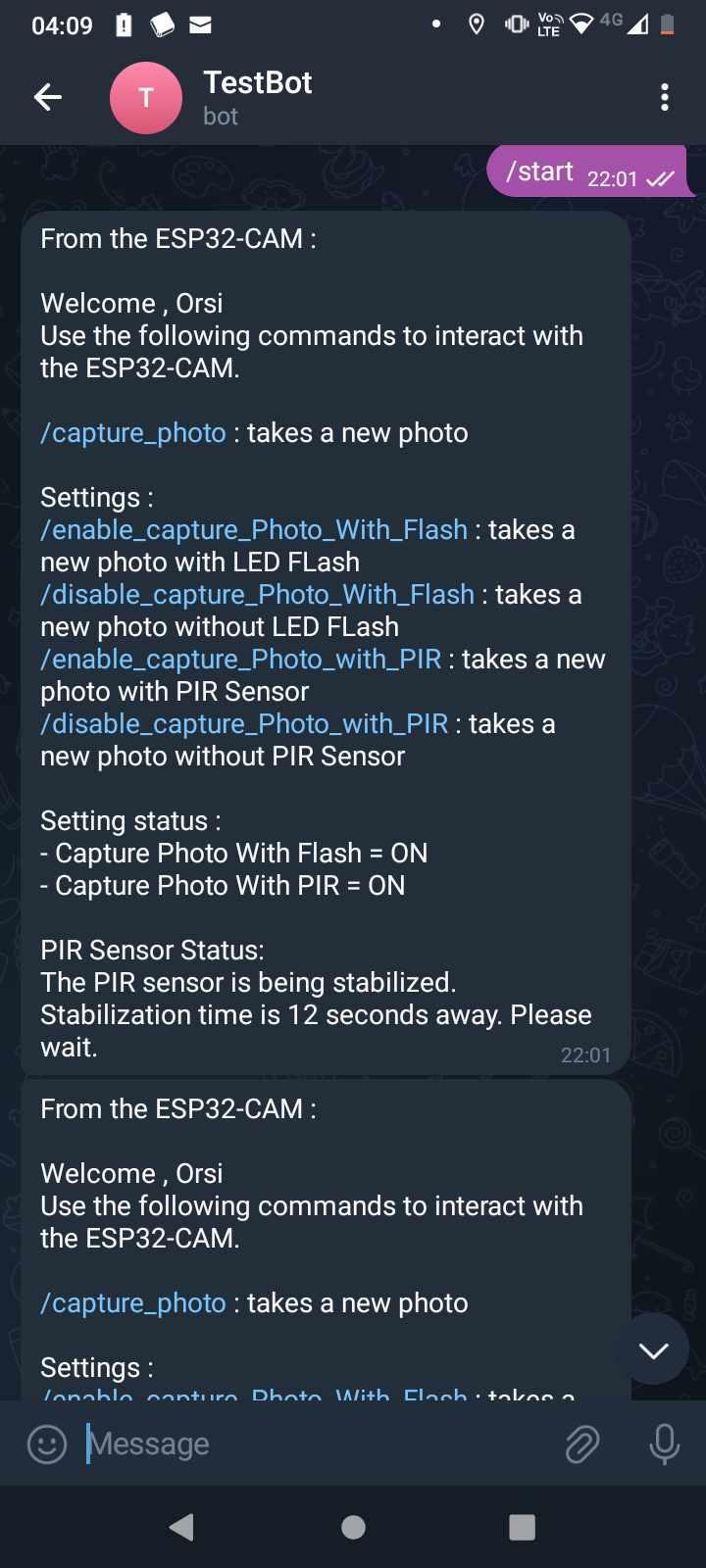
*bool capture\_Photo\_with\_PIR\_state();*

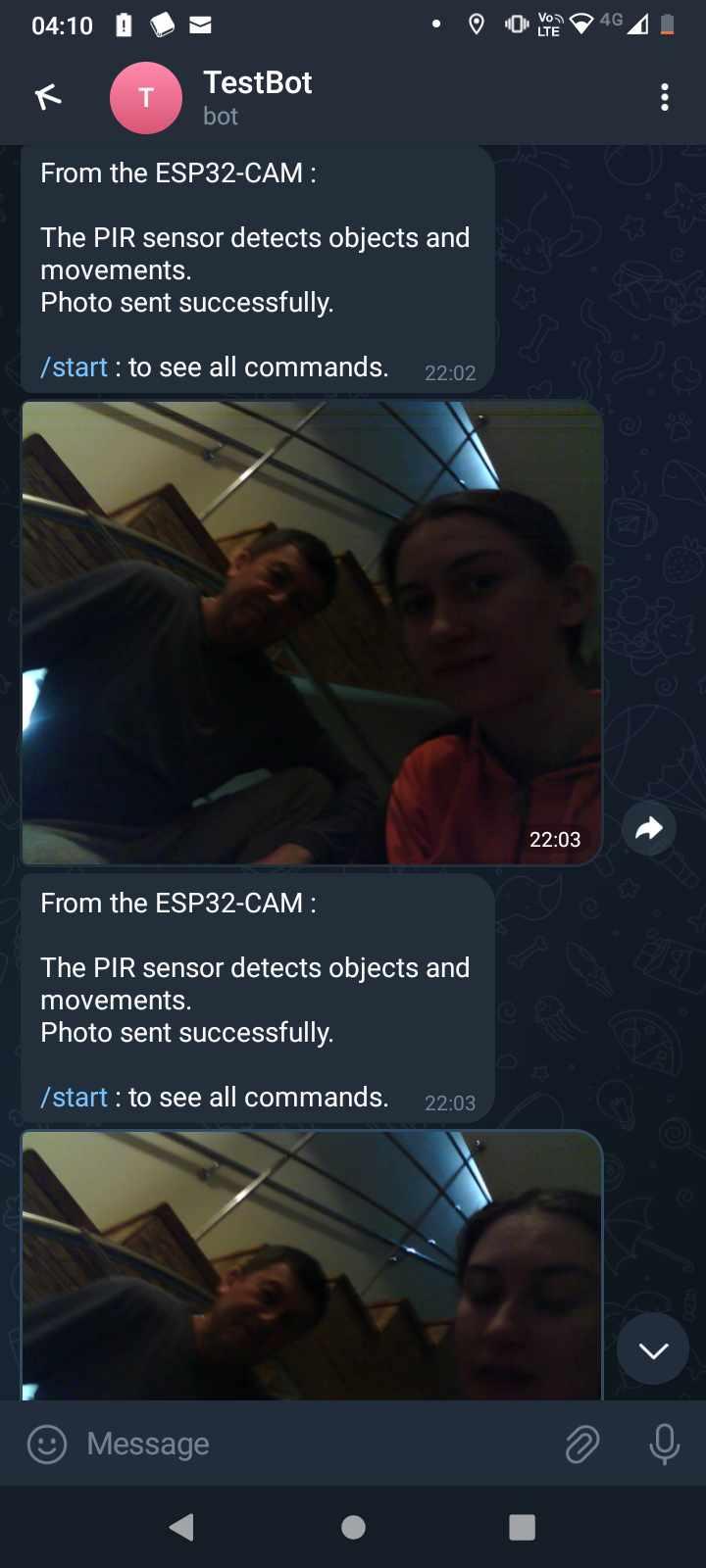
## Demonstráció

#### 

#### 

#### 





## Könyvészet

[1] <https://randomnerdtutorials.com/telegram-esp32-cam-photo-arduino/>

[2] <https://github.com/espressif/arduino-esp32>

[3] <https://lastminuteengineers.com/getting-started-with-esp32-cam/>

[4] <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

[5] <https://randomnerdtutorials.com/esp32-cam-pir-motion-detector-photo-capture/>