

# IDOM

## Instrukcja użytkownika

### Moduł kamery

#### Spis treści

1	Zastosowanie	2
2	Potrzebne elementy	2
3	Schemat połączeń	2
4	Programowanie modułu	3

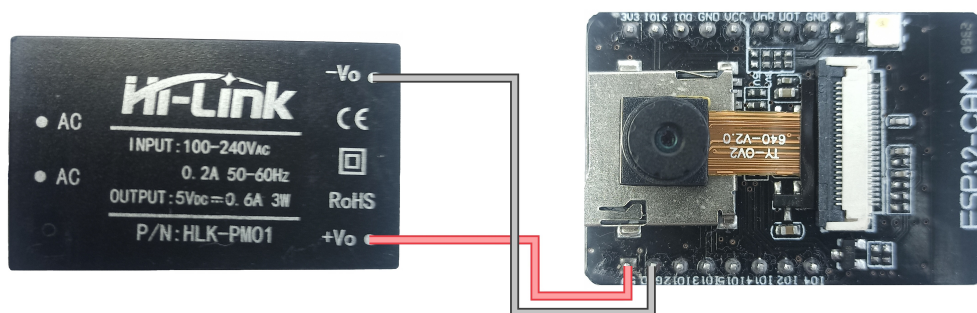
## 1 Zastosowanie

Moduł kamery daje nam podgląd na wybrane przez nas miejsce. Kamera wysyła 10 klatek na sekundę w rozmiarze 2Mpix oraz posiada funkcję wykrywania twarzy.

## 2 Potrzebne elementy

Nazwa	ilość
Zasilacz Hi-Link AC 100-240V To DC 5V	1
ESP32-CAM	1
Przewód połączeniowy żeńsko-żeński	2
Sznur przyłączeniowy z wtyczką	1

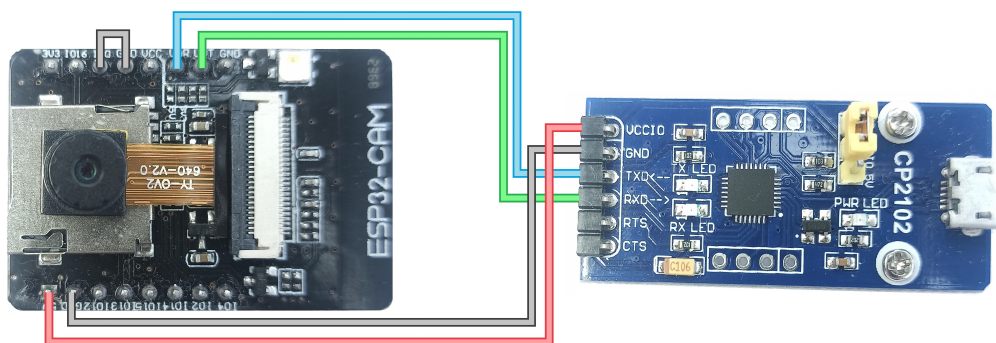
## 3 Schemat połączeń



## 4 Programowanie modułu

Moduł kamery jest jedynym, który nie używa ESP8266 tylko ESP32. Jest to spowodowane większą mocą obliczeniową ESP32 oraz dostępnością płytki dostosowanej do obsługi kamery. Dla ESP32 musimy dodać dodatkowy URL do menedżera płytek. W tym celu otwieramy Plik » Preferencje i w polu "Dodatkowe adresy URL do menedżera płytek:" wstawiamy przecinek i wklejamy ten link [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json).. Teraz możemy przejść do Narzędzia » Płytki: » Menedżer płytek... Wskoczy nam nowe okno, w którym wpisujemy esp32 i instalujemy płytkę, która nam się pokazała. Po instalacji wyłączamy i ponownie włączamy program Arduino IDE. Po otwarciu programu przechodzimy do Narzędzia » Płytki » ESP32 Arduino i wybieramy "ESP32 Wrover Module".

Aby zaprogramować ESP32-CAM musimy podłączyć je zgodnie ze schematem



Teraz możemy wgrać oprogramowanie na nasze ESP. Do zaprogramowania należy użyć pliku [Camera.ino](#). Pomimo użycia ESP32 a nie ESP8266 musimy pamiętać o 2 bibliotekach z poradnika "Wgrywanie oprogramowania na mikrokontrolery ESP".

Na górze pliku znajduje się kilka zmiennych do ustawienia

```
/***** Variables to customize *****/
```

Najpierw ustawiamy nazwę naszej sieci (musi to być sieć 2.4GHz)

```
// Name of your network  
const char* ssid = "";
```

Następnie ustawiamy hasło naszej sieci

```
// Password of your network  
const char* password = "";
```

w tym miejscu wpisujemy adres naszego serwera razem z dopisanym portem  
8001 na przykład "192.168.0.10:8001"

```
// Raspberry server address here  
const String ServerName = "";
```

Ustawiamy nazwę sensora na taką samą jak w aplikacji IDOM

```
// Name of your sensor  
String Name = "";
```

Po ustawieniu tych wartości możemy wgrać kod a następnie cieszyć się działającą kamerą.