

---

## Temat projektu i założenia

---

Głównym założeniem projektu było skonstruowanie urządzenia, które będzie pomagało w zdalnej kontroli domku dla zwierząt. Układ odpowiedzialny będzie za mechaniczne sterowanie wejściem (otwieraniem/zamykaniem wejścia/wyjścia domku dla zwierząt). Pozwoli to na niesamowicie wygodne kontrolowanie wejścia/wyjścia do domku dla zwierząt. Projekt w swojej funkcjonalności oferował będzie również możliwość zliczania zwierząt (obiektów), dzięki czemu możliwe będzie uzyskanie ilości zwierząt znajdujących się już w środku. Takie połączenie układu umożliwi w przyszłości (poza projektem z przedmiotu) jego rozbudowę od strony programowej. Całość kontrolowana będzie z poziomu serwera http.

---

## Analiza zadania

---

### Moduł procesora ESP32:

Moduł procesora				
Numer katalogowy	Napięcie zasilania	Wifi	Protokół Bluetooth	Cena
<b>ESP32 ESP-WROOM-32U</b>	5V/3,3V	2,4 GHz	4.2 Low Energy	35,34zł
<b>DFR-21227</b>	3,3V	2,4 GHz	5.0 Low Energy	45,9zł
<b>MOD-08893</b>	5V	2,4 GHz	4.2 Low Energy	44,7zł

Moduł **ESP32 ESP-WROOM-32U** wybrano ze względu na przecenę związaną z Black Friday oraz możliwość dołączenia do niego zewnętrznej anteny poprzez złącze U.FL. Wybrano tańszy moduł ze względu na bardzo podobne parametry modułów. Ilość wyjść GPIO, nie miała zbyt dużego znaczenia podczas doboru procesora, gdyż projektowany układ jest bardzo prosty. Umożliwi to w przyszłości jedynie rozbudowę jego funkcjonalności.

Wykorzystana zostanie również dodatkowa antena, z powodu umiejscowienia domku na obrzeżu sygnału wifi. Pozwoli to na bezproblemowe obsługiwanie mikrokontrolera. Antena WiFi 3dBi do ESP32 o kodzie producenta: 5904501663090.

Specyfikacja anteny:

- Wzmocnienie: 3 dBi
- Częstotliwość pracy: 2.4 GHz
- Antena łamana
- Długość przewodu: 15 cm

## Czujnik przerwania wiązki:

Czujnik przerwania wiązki				
Numer katalogowy	Wielkość diody	Zasięg	Dostępność	Cena
<b>DNG-18687</b>	3mm	25cm	Dobra	12,4zł
<b>DNG-18689</b>	3mm	50cm	Wysoka	14zł
<b>DNG-18688</b>	5mm	25cm	Niska	13,9zł
<b>DNG-18690</b>	5mm	50cm	Niedostępny	12,9zł

Wybrano czujnik **DNG-18688**, ze względu na dość wąskie przejście dla zwierząt. W projekcie najprawdopodobniej zostaną użyte dwa czujniki przerwania wiązki, aby móc określić kierunek przerwania. Do zliczania obiektów można by użyć również czujnik mikrofalowy, który również może zostać wzięty pod uwagę, w zależności dostępu czasowego.

## Moduł mostka-h:

Moduł mostka-h				
Numer katalogowy	Napięcie sterowania	Zakres pracy napięcie	Maksymalny prąd	Cena
<b>BTS7960B</b>	5V	5V-27V	43A	40zł
<b>DC L298N</b>	5V	3,3V-46V	2A	15,23zł
<b>TA6586</b>	2.2-6V	3-14V	5A	18,35zł

Wybór był oczywisty i padł na moduł **TA6586**. Niestety na rynku nie znalazłem mostka-h w niższej cenie, pozwalającego na sterowanie silnikiem pod obciążeniem prądu w wysokości 5A. Większość modułów jest niestety niedostępna, lub czas oczekiwania na nie jest zbyt długi. Wybrany mostek bardzo trafnie spełnił wymagania co do użytkowania.

## Siłownik elektryczny:

Na potrzeby domku wymagalny jest minimalny wysuw wynoszący 20 cm. Załączone siłowniki posiadają wbudowane czujniki krańcowe, umożliwiające zatrzymanie pracy silnika.

Siłownik elektryczny				
Numer katalogowy	Wysuw	Napięcie zasilania	Maksymalny pobór prądu	Cena
<b>MOT-09875</b>	25cm	12V	1,5A	199zł
<b>MOT-12672</b>	20cm	12V	1,5A	175zł
<b>ELB-16009</b>	25cm	24V	2,1A	199zł
<b>MOT-05363</b>	25cm	12V	4,2A	199zł
<b>MOT-05365</b>	44cm	12V	4,2A	219zł

Wybrano siłownik o numerze: **MOT-05365**, ze względu na większy wysuw, który pozwoli na szersze otwarcie drzwi do domku dla zwierząt. Siłownik będzie osobno zasilany zasilaczem 12V, 5A. Jego polaryzację będzie kontrolował wyżej wybrany mostek-h.

Program został napisany w języku C++ w środowisku Visual Studio Code z zainstalowanym rozszerzeniem PlatformIO, które posiada wiele przydatnych funkcjonalności m.in. import bibliotek jest bardzo prosty, dzięki rozbudowanej wyszukiwarce bibliotek, lub poprzez manualne dodawanie biblioteki do folderu Lib. Sprawia to, że pisanie programu staje się dużo szybsze, szczególnie że programista nie musi w ten sposób pisać własnych bibliotek do sterowania poszczególnymi elementami.

---

## Opis zmian w przesłanej analizie

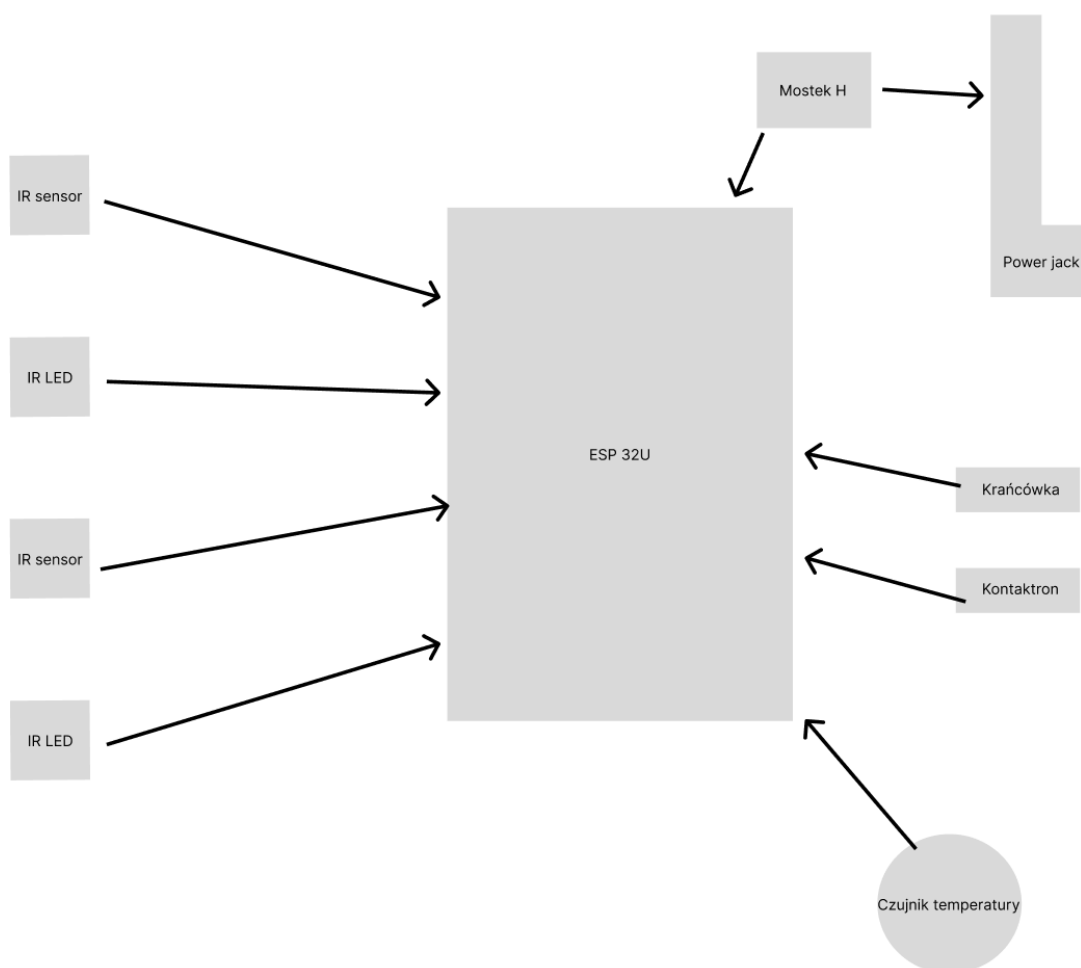
---

Początkowo wybrany został mostek **BTS7960B**, lecz po dostawie okazało się, że jest wadliwy i nie działa. W międzyczasie ponownie dostępny został **TA6586**. Pozwoliło to zaoszczędzić trochę pieniędzy oraz zmniejszyć rozmiar projektu.

---

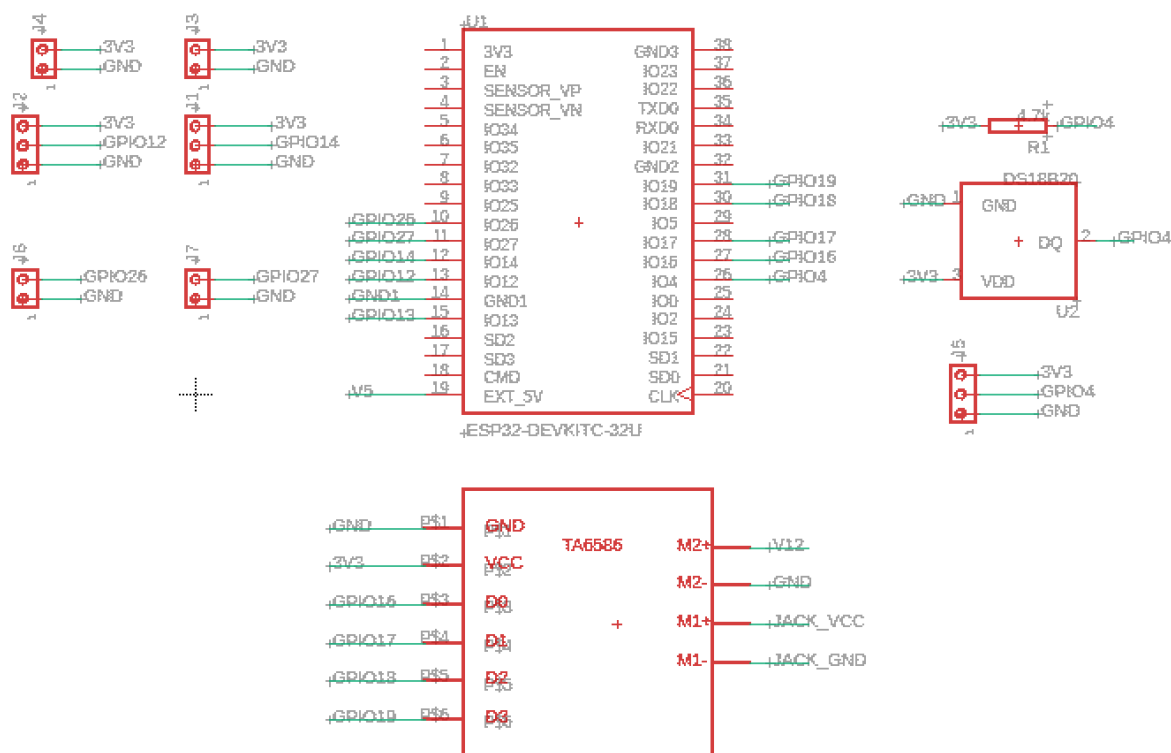
## Specyfikacja wewnętrzna urządzenia

---



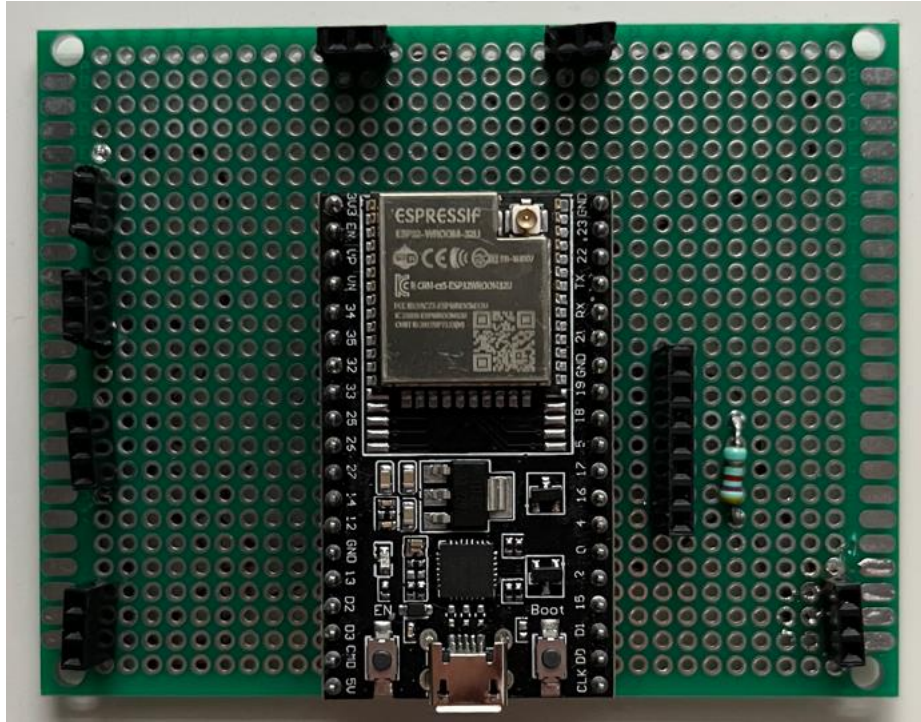
Rysunek 1 Schemat blokowy urządzenia

ESP 32U – mikrokontroler, steruje całym urządzeniem,  
 IR LED – dioda emitująca światło podczerwone,  
 IR sensor – czujnik światła podczerwonego, dzięki któremu możliwe jest wykrycie przerwania wiązki świetlnej,  
 Power jack – siłownik sterowany elektrycznie, pozwalający na otwieranie oraz zamykanie domku,  
 Mostek H – sterowany przez mikrokontroler zmienia polaryzację silnika, co zmienia kierunek obrotu silnika,  
 Krańcówka – została użyta do sygnalizacji w pełni otwartych drzwi do domku  
 Kontaktron – został użyty do sygnalizacji w pełni zamkniętych drzwi do domku,  
 Czujnik temperatury – służy do pomiaru temperatury oraz przekazuje dane do mikrokontrolera.



## Opis montażu układu:

Układ został zlutowany na uniwersalnej płytce prototypowej, z zachowaniem szczególnego bezpieczeństwa.



*Zdjęcie 1 Zlutowany projekt na płytce PCB*

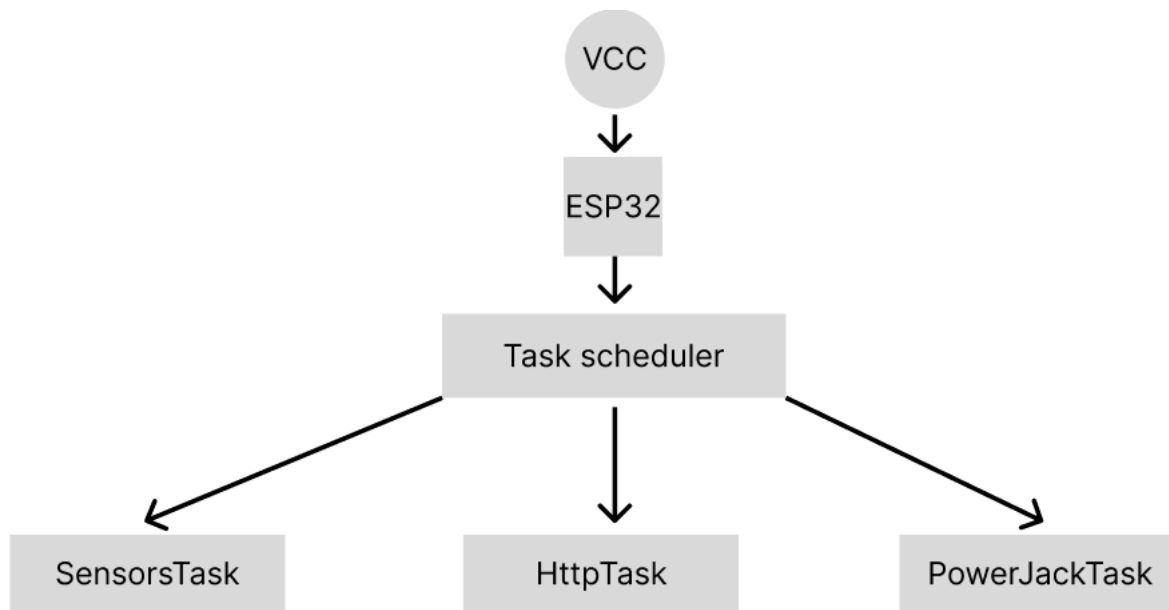
## Opis sposobu programowania układu:

Mikrokontroler programowany jest przy użyciu skryptu napisanego w pythonie, dostarczonego od firmy espressif poprzez interfejs micro USB. W celu zaprogramowania układu potrzebne jest wprowadzenie mikrokontrolera w tzw. „Download Mode”, wówczas procesor pozwala na dostęp do pamięci procesora.

## Opis sposobu uruchamiania oraz testowania:

Układ uruchamiany jest automatycznie po podpięciu zasilania, na samym początku mikrokontroler próbuje połączyć się z siecią oraz po połączeniu włączyć asynchroniczny serwer na urządzeniu. W przypadku braku połączenia z Internetem (większa nieudana ilość błędnych połączeń), mikrokontroler się zrestartuje. Przewidziano również utratę połączenia z siecią, dlatego co jakiś czas pingowany jest adres google.com, aby sprawdzić połączenie z siecią. Jeżeli procesor nie otrzyma odpowiedzi na pakiety ICMP, zostanie zrestartowany.

## Podział tasków oraz opis procedur:



*Zdjęcie 2 Podział tasków w programie*

SensorsTask() – zajmuje się głównie odczytem danych z wszystkich czujników:

- Przedni czujnik przerwania,
- Tylni czujnik przerwania,
- Czujnik temperatury,
- Krańcówka,
- Kontaktron.

Głównym zadaniem jest poprawne ustawienie zmiennych globalnych, używanych później w programie.

```
void vSensorsTask(void* parameters)
{
    // Pin configuration
    Serial.println("Starting SensorsTask...");
    pinMode(PIN_IR_FRONT_SENSOR, INPUT);
    digitalWrite(PIN_IR_FRONT_SENSOR, HIGH);    // Pull-up resistor

    #ifdef DEBUG_SENSORS
    Serial.printf("[%u] Configured IR Reciever GPIO: %i\n", xTaskGetTickCount, PIN_IR_FRONT_SENSOR);
    #endif

    pinMode(PIN_IR_REAR_SENSOR, INPUT);
    digitalWrite(PIN_IR_REAR_SENSOR, HIGH);    // Pull-up resistor

    #ifdef DEBUG_SENSORS
    Serial.printf("[%u] Configured IR Reciever GPIO: %i\n", xTaskGetTickCount, PIN_IR_REAR_SENSOR);
    #endif

    // States 00 if front 01 if rear
    uint8_t previousInterrupt = -1;
    bool wentThrough = true;

    // Setup a oneWire instance to communicate with any OneWire devices
    OneWire oneWire(PIN_DS18B20_TEMPERATURE_SENSOR);
    // Pass our oneWire reference to Dallas Temperature sensor
    DallasTemperature dallasTempSensors(&oneWire);
    dallasTempSensors.begin();

    // Door sensors
    pinMode(PIN_FRONT_DOOR_SENSOR, INPUT_PULLUP);
    pinMode(PIN_BACK_DOOR_SENSOR, INPUT_PULLUP);
}
```

*Zdjęcie 3 Przykład konfiguracji czujników*



```
String sendHTML(){
String ptr = "<!DOCTYPE html> <html>\n";
ptr += "<title>Smart Domek</title>\n";
ptr += "<meta charset='UTF-8' name='viewport' content='width=device-width, initial-scale=1'>\n";
ptr += "</head><body bgcolor = '#434242'>\n";
ptr += "<div id='information' style='color:#22A39F; font-family: Georgia; font-size: 40px; margin: auto; text-align: center;'>\n";
ptr += "<h4>Smart Domek</h4>\n";
ptr += "</div>\n";
ptr += "<div id = 'temperature' style='color:#22A39F; font-family: Georgia; font-size: 20px; margin: auto; text-align: center; margin-top: 30px'>\n";
ptr += "<div id = 'temperatureReading' style='color:#F3EFE0; font-family: Georgia; font-size: 25px; margin: auto; text-align: center;'><span id = 'temp'>\n";
ptr += "<div id = 'temperature' style='color:#22A39F; font-family: Georgia; font-size: 20px; margin: auto; text-align: center; margin-top: 30px'>\n";
ptr += "<div id = 'animalsInside' style='color:#F3EFE0; font-family: Georgia; font-size: 25px; margin: auto; text-align: center;'><span id = 'animals'>\n";
ptr += "<br>\n";
ptr += "<form action='/get' method='GET' style='display: flex; align-items: center; justify-content: center;'>\n";
ptr += "<input type='button' value='OFF' style='font-family: Georgia; margin: 12px; margin-top: auto; padding: 5px 15px;'>";
ptr += "<input type='button' value='ON' style='font-family: Georgia; margin: 12px; margin-top: auto; padding: 5px 15px;'>";
ptr += "</form>\n";
ptr += "<br>\n";
ptr += "<div id='xd' style='display: flex; margin: auto; text-align: center; ; justify-content: center; align-items: center;'>\n";
ptr += "<form action='/get' method='GET'>\n";
ptr += "<input type='button' value='OPEN' style='font-family: Georgia; background-color:#22A39F; border: none;border-radius: 10px;'>";
ptr += "<input type='button' value='CLOSE' style='font-family: Georgia; background-color:#22A39F; border: none;border-radius: 10px;'>\n";
ptr += "</form>\n";
ptr += "</div>\n";

if(doorError)
{
ptr += "<div id='doorError' style='color:red; font-family: Georgia; font-size: 32px; margin: auto; text-align: center;'>Door Error!!!</div>\n";
}

// -----JavaScript----->
ptr+= "<script>setInterval(function() {getSensorData(); getAnimalCounter();}, 10);function getSensorData() {var xhttp = new XMLHttpRequest();xhttp.open('GET', '/get', true);xhttp.send();}\n";
ptr += "</body>\n";
ptr += "</html>\n";

return ptr;
}
```

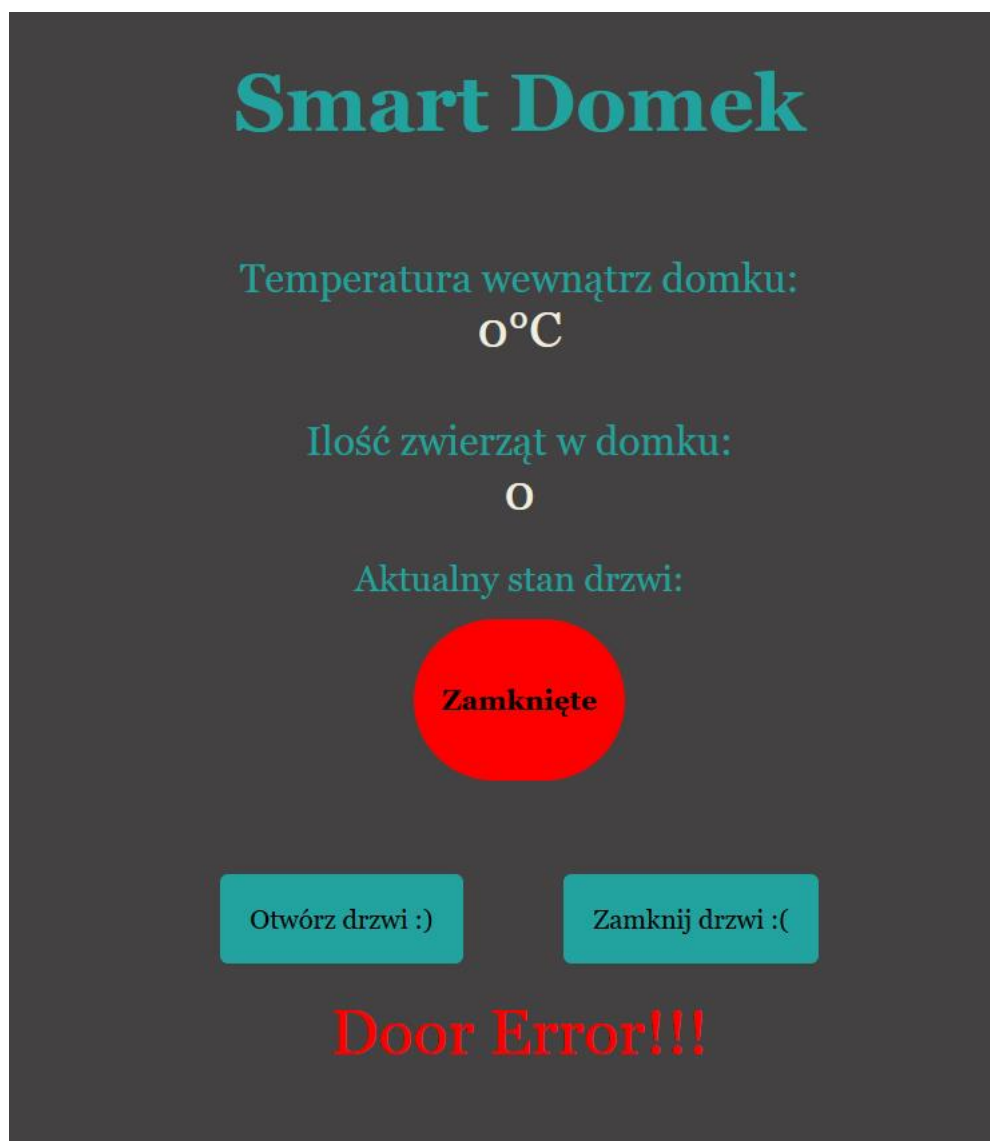
PowerJackTask() – odpowiada za sterowanie pracą silnika podczas wykonywania programu.

```
static uint32_t frontIRSensorState; - przerwanie przedniego czujnika
static uint32_t rearIRSensorState; - przerwanie tylnego czujnika
extern uint32_t numberOfAnimalsInside; - ilość zwierząt w domku
extern float temperatureReadingC; - odczyt temperatury wewnątrz
```

---

## Interfejs użytkownika

---



*Zdjęcie 5 Wygląd strony przez użytkownika oraz obsługa błędnego stanu*

Obsługa całego programu odbywa się z poziomu serwera http udostępnionego przez mikrokontroler ESP32. W sieci lokalnej dodany został DNS na routerze, pozwalający połączenie się po wygodniejszym po stronie użytkownika adresie smartdomek.pl. Jeżeli z jakiegokolwiek powodu nie nastąpi sygnalizacja (poprzez przekaźnik, bądź kontaktron), to serwer zgłosi błąd na stronie http. Użytkownik po poinformowaniu może zareagować i manualnie naprawić problem, bądź znaleźć jego przyczynę.