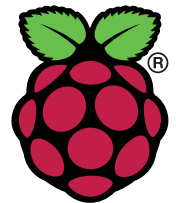


Laserowy system zabezpieczeń na Raspberry Pi



Istnieje wiele sensorów, przeznaczonych do detekcji wtargnięcia w systemach alarmowych, jednym z nich są bariery optyczne. W artykule zaprezentowano barierę laserową, wyposażoną w komputer jednopłytkowy Raspberry Pi, która zaalarmuje użytkownika poprzez e-mail oraz udostępni strumień wideo z nadzorowanego miejsca.

Projekt składa się z trzech osobnych podsystemów: komputera jednopłytkowego Raspberry Pi, laserowej bariery optycznej i komputera do zdalnej obsługi i monitorowania systemu. Celem projektu jest wykrywanie wtargnięcia intruza na chroniony teren, a następnie zaalarmowanie obserwującego system użytkownika oraz udostępnienie podglądu nadzorowanego miejsca.

Bariery optyczne (laserowe i inne) doskonale sprawdzają się do monitorowania nawet dużego terenu, gdyż nie wymagają skomplikowanej instalacji – wymagane są tylko dwa punkty, które nawet nie muszą być połączone ze sobą elektrycznie. Taki system może się sprawdzić np. jako detektor monitorujący teren zielony, ogród czy podwórze.

Aby zrealizować projekt, nie trzeba mieć ogromnej wiedzy elektronicznej, konieczna jest znajomość podstaw elektroniki i posługiwania się lutownicą. Wartościowa będzie natomiast umiejętność z zakresu programowania w Pythonie, w szczególności na Raspberry Pi i obsługa tego komputera jednopłytkowego z poziomu terminalu.

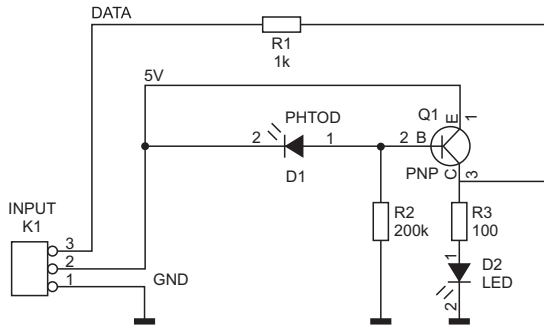
Potrzebne komponenty

Aby zbudować opisane poniżej urządzenie, potrzebne są następujące elementy:

- Co najmniej jeden działający komputer jednopłytkowy Raspberry Pi. Autor projektu użył pierwszego modelu w wersji B z zainstalowanym Raspbianem Wheezy. Projekt będzie działał również dobrze z innymi wersjami Raspbiana;
- Zewnętrzne źródło zasilania dla Raspberry Pi. Układ może pobierać dość duży prąd podczas pracy, w związku z czym potrzebny jest zasilacz o odpowiedniej mocy. Zbyt niska moc zasilacza może powodować ciężkie do debugowania problemy z Raspberry Pi;
- Wskaźnik lub dioda laserowa. Nie ma wielkich wymagań co do lasera, wystarczy tutaj prosty i tani czerwony wskaźnik laserowy. Interesuje nas taki, którego długość emitowanej fali wynosi w przybliżeniu 630...680 nm;
- Kamera na USB. Dowolna kamerka współpracująca z Raspberry Pi będzie odpowiednia. Warto wybrać taką, która będzie łatwa w montażu i integracji z systemem.

Dodatkowo potrzebne będą:

- płytki uniwersalna lub płytki stykowa, na której wykonamy układy,
- dioda LED,
- oporniki o rezystancji 100 Ω , 1 k Ω , 10 k Ω i zawierający się w zakresie od 200 do 1000 k Ω ,
- tranzystor 2N4403 lub zamiennik,



Rysunek 1. Schemat sensora promienia lasera

- fotodiody odpowiednia do pasma pracy lasera, na przykład PDB-C142-ND.

Układ można również zmontować na odpowiedniej płycie drukowanej (pliki projektowe dostępne są na stronie z projektem), ale z uwagi na prostotę układu nie jest to konieczne.

Oprócz powyższych materiałów przydadzą się także pewne narzędzia (fizyczne i programistyczne):

- Hub USB, jeśli chcemy do Raspberry Pi podłączyć większą liczbę urządzeń (np. kamer lub kamerę, klawiaturę i mysz);
- Multimetr lub oscyloskop (najlepiej oba);
- Synergy (<https://bit.ly/3dq8hYK>) zainstalowane na Raspberry Pi, które pozwala kontrolować mysz i klawiaturę Raspberry Pi za pomocą innego, zdalnego komputera. Oprogramowanie to sporo upraszcza jeżeli chodzi o obsługę komputera, zwłaszcza jeżeli preferujemy zdalną konfigurację lub po prostu nie przepadamy za pracą bezpośrednio na Raspberry Pi;
- MobaXterm (<https://bit.ly/3dq7abe>), również zainstalowane na Raspberry Pi – pozwoli na zdalne sterowanie komputerem i manipulowanie plikami poprzez SSH.

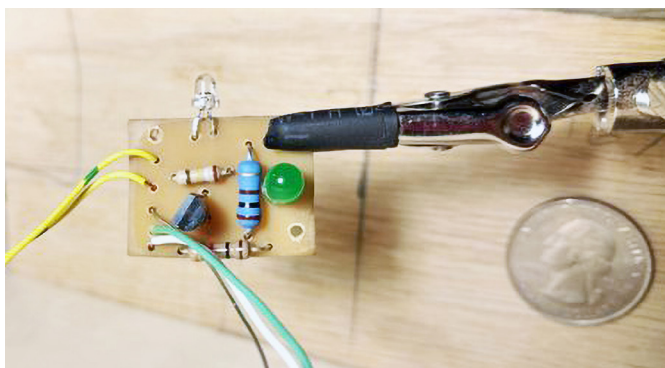
Uruchomienie Raspberry Pi

Zanim rozpoczniemy testy z laserem, trzeba przebić się przez kilka warstw oprogramowania. Dobrą praktyką jest zacząć od czystego systemu operacyjnego, zatem zainstaluj na karcie nowy obraz systemu operacyjnego.

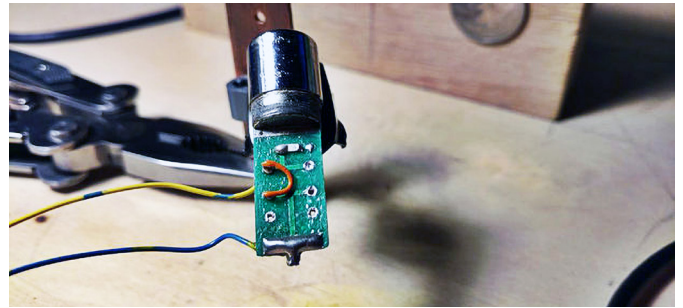
Nawet jeśli nie jest możliwe zainstalowanie systemu operacyjnego od nowa, należy upewnić się, że Raspbian, jaki mamy na pokładzie komputera jednopłytkowego, jest w najnowszej wersji. Aby zaktualizować system i zainstalowane pakiety w terminalu, należy wpisać: `sudo apt-get update`

Laserowy system detekcji

Przystępujemy do montażu detektora promienia laserowego. Układ można zmontować na płycie uniwersalnej lub płycie stykowej. Na **rysunku 1** pokazano schemat detektora. Jest on zasilany napięciem 5 V, które może być pobierane z zewnętrznego zasilacza lub bezpośrednio z pinów złącza GPIO Raspberry Pi. Zastosowano fotodiody jako przełącznik do wyzwolenia tranzystora. Tranzystor wzmacnia niewielką



Fotografia 1. Zdjęcie zmontowanego sensora promienia lasera



Fotografia 2. Dolutowana zwora, zastępująca przycisk chwilowy w module lasera

zmianę napięcia na fotodiody na sygnał cyfrowy, który Raspberry Pi może odczytać poprzez pin GPIO.

Komputer jednopłytkowy pobiera sygnał z punktu pomiędzy rezystorem 100 Ω a diodą LED, sygnalizującą stan układu. Na **fotografii 1** pokazano zmontowany sensor.

Modyfikacja lasera do zasilania zewnętrznego

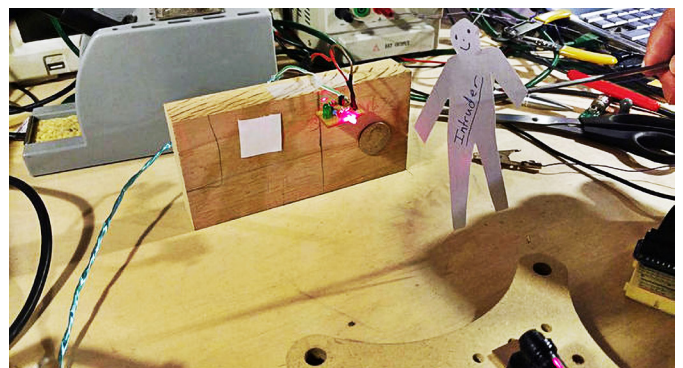
Kolejnym krokiem jest zbudowanie źródła światła dla bariery laserowej. Zastosujemy tutaj komercyjny wskaźnik laserowy. Konieczne są jednak pewne modyfikacje tego urządzenia, by możliwe było zasilanie z zewnętrznego zasilacza. Należy wykonać następujące czynności:

1. Rozbierz wskaźnik, wyjmując moduł laserowy (wraz z optyką) z obudowy urządzenia. Robiąc to, należy zwrócić uwagę na ustawienie zacisków akumulatora względem elektroniki lasera.
2. Odłutuj sprężynę, łączącą akumulator i przycisk chwilowy wskaźnika.
3. Pomiędzy punktami, gdzie znajdował się przycisk, wlotuj przewód. W ten sposób, gdy zasilanie lasera zostanie włączone, automatycznie się on załączy (**fotografia 2**).
4. Przylutuj przewody do dodatniego i ujemnego bieguna zasilania modułu laserowego. Zależnie od konstrukcji lasera punkty te mogą być w różnych miejscach, dlatego też należy dokładnie prześledzić ścieżki zasilania i zdobyć się na odrobinę inżynierii wstecznej. Po poprawnym podłączeniu zasilania laser powinien od razu zaświecić.
5. Ostatnim krokiem jest umieszczenie lasera na słupku montażowym, co pozwoli z łatwością pozycjonować go w gotowym systemie, by wycelować w odbiornik.
6. Finalnie dobrze jest zabezpieczyć elektronikę lasera np. klejem na ciepło i odpowiednią obudową.

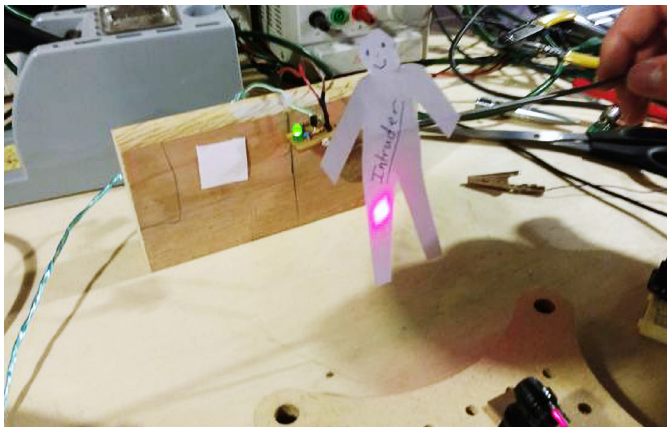
Testowanie systemu

Możemy teraz przetestować barierę laserową. W pierwszej kolejności należy podłączyć laser do zasilania (5 V) np. na płytce stykowej, którą dołączymy do Raspberry Pi lub zasilacza USB. Laser, po podaniu zasilania, powinien od razu zacząć emitować. Jeśli nie, sprawdźmy wykonane wcześniej połączenia i poprawmy ewentualne błędy.

Teraz możemy przetestować obwód detekcji. Skierujmy laser na fotodiody, jak pokazano na **fotografii 3**. Dioda LED powinna zgasnąć,



Fotografia 3. Testowo zestawiona bariera laserowa



Fotografia 4. Test bariery laserowej – laser zastąpiony

sygnalizując, że wiązka jest aktywna. W momencie, gdy przesłoniemy wiązkę laserową, jak na **fotografii 4**, dioda LED na module detektora powinna ponownie się zapalić.

Podłączenie kamery internetowej do Raspberry Pi

Po uruchomieniu optycznej zapor laserowej możemy podłączyć do Raspberry Pi kamerę na USB, która będzie uruchamiana sygnałem z bariery laserowej. Rozpoczynamy od podłączenia kamery do portu USB. System powinien automatycznie wykryć i zainstalować kamerę wraz z potrzebnymi sterownikami. My musimy tylko doinstalować narzędzie MJPG-Streamer, które służy do strumieniowego przesyłania wideo. W tym celu musimy pobrać kod źródłowy i zbudować całą aplikację, postępując zgodnie z dalszym opisem.

W pierwszej kolejności musimy pobrać i zainstalować zależności, które wymagane będą podczas kompilacji i działania MJPG-streamera. W tym celu w terminalu wpisujemy:

```
sudo apt-get install libjpeg8-dev imagemagick libv4l-dev
```

Następnie musimy dodać brakujący plik *videodev.h*. Dostępny jest on w systemie pod zmienioną nazwą (*videodev2.h*), więc aby usatysfakcjonować budowany pakiet, wystarczy stworzyć dowiązanie symboliczne:

```
sudo ln -s /usr/include/linux/videodev2.h /usr/include/linux/videodev.h
```

Finalnie możemy pobrać kody źródłowe. Możemy to zrobić, pobierając bezpośrednio spakowaną paczkę z kodem:

```
cd ~
wget http://sourceforge.net/code-snapshots/svn/m/mj/mjpg-streamer/code/mjpg-streamer-code-182.zip
```

Paczkę można pobrać też m.in. przez przeglądarkę spod linku <https://bit.ly/36dyczM>. Następnie należy ją rozpakować:

```
unzip mjpg-streamer-code-182.zip
cd mjpg-streamer-code-182/mjpg-streamer
```

Możemy także ściągnąć ją bezpośrednio z repozytorium SVN. W tym celu musimy zainstalować klienta Subversion, jeśli go jeszcze nie mamy:

```
sudo apt-get install subversion
```

I wtedy możemy pobrać aktualne kody źródłowe:

```
cd ~
```

```
svn co https://svn.code.sf.net/p/mjpg-streamer/code/mjpg-streamer/ mjpg-streamer
cd mjpg-streamer
```

Niezależnie od wybranego sposobu mamy teraz kod źródłowy aplikacji, który możemy zbudować. W tym celu wpisujemy w terminalu (będąc w folderze z pobranym kodem):

```
make mjpg_streamer input_file.so output_http.so
```

Po zbudowaniu aplikacji możemy ją zainstalować. Poniższe komendy kopiują wszystkie potrzebne pliki do folderów systemowych:

```
sudo cp mjpg_streamer /usr/local/bin
sudo cp output_http.so input_file.so /usr/local/lib/
sudo cp -R www /usr/local/www
```

Teraz możemy sprawdzić, czy kamera i program do przesyłania strumienia wideo działają. W pierwszej kolejności uruchamiamy kamerę:

```
mkdir /tmp/stream
raspistill --nopreview -w 640 -h 480 -q 5 -o /tmp/stream/pic.jpg -tl 100 -t 9999999 -th 0:0:0 &
```

A następnie MJPG-Streamer – kamera już zapisuje obrazy, więc wystarczy uruchomić strumień wideo:

```
LD_LIBRARY_PATH=/usr/local/lib mjpg_streamer -i „input_file.so -f /tmp/stream -n pic.jpg” -o „output_http.so -w /usr/local/www”
```

Aby obejrzeć strumień wideo, trzeba wejść na stronę <http://localhost:8080>, jeżeli oglądamy wideo z poziomu Raspberry Pi, albo <http://<numer-IP-Raspberry Pi>:8080>, jeśli strumień chcemy oglądać ze zdalnej maszyny.

Po zakończonej instalacji pozostaje nam tylko posprzątać, to jest usunąć wykorzystany do kompilacji kod źródłowy.

```
cd ../../
rm -rf mjpg-streamer-182
```

Lub skasować go z innego miejsca, jeśli pobieraliśmy źródło z innej lokalizacji lub korzystaliśmy z SVN-a.

Finalnie uruchamiamy MJPG-streamer w docelowej konfiguracji: `mjpg_streamer -i „./input_uvc.so -y /dev/video0” -o „./output_http.so -w ./www -p 8081”`

To polecenie skonfiguruje i uruchomi strumień wideo na porcie 8081 przy użyciu źródła *video0*. Jeśli chcesz uruchomić wiele kamer internetowych, wprowadź ponownie powyższe polecenie, ale zmień *video0* na np. *video1* i *8081* na *8082*. W ten sposób kolejne kamery transmitować będą strumień na kolejnych portach. Oczywiście nazwy urządzeń w systemie można zmienić, według stanu na naszym urządzeniu, a port również dobrać można do własnych potrzeb. Stan działania systemu sprawdzamy tak jak powyżej, wchodząc na stronę www.

Integracja sprzętu

Teraz można podłączyć do Raspberry Pi wszystkie elementy. Na razie laser zasilamy z Raspberry Pi, ale docelowo możliwe jest jego zewnętrzne zasilanie, zwłaszcza że może być umieszczony dość daleko od detektora z komputerem. Do Raspberry Pi podłączamy także detektor. Wyjście sensora laserowego podpinamy do pinu 25 portu GPIO komputera jednopłytkowego. Dodatkowo, musimy ściągnąć tę linię opornikiem do masy. Można to zrealizować z pomocą rezystora 10 kΩ, ale każdy inny o podobnej rezystancji będzie dobry. Teraz można przystąpić do programowego wykrywania zmiany stanu laserowej bariery optycznej.

REKLAMA

Prenumerujesz Elektronikę Praktyczną?

Zaloguj się na swoje konto Prenumeratora (www.avt.pl/uzytkownik)

i pobierz ZA DARMO e-wydania kilkudziesięciu czasopism z serii **#CzasNaCzytanie**

Prenumeratę zamówisz na www.avt.pl/prenumerata



Programowe monitorowanie lasera

Do przetestowania bariery optycznej wystarczy nam prosty program, który monitoruje stan linii GPIO i podaje informacje o zmianach w terminalu. Skrypt ten napisany jest w Pythonie, który potrzebuje do obsługi linii GPIO odpowiednich bibliotek. Aby je zainstalować, w linii komend wpisujemy: `sudo apt-get install python-rpi-gpio python3-rpi-gpio`

Następnie tworzymy plik `read_pin.py` i nadajemy odpowiednie uprawnienia, by możliwe było jego uruchomienie. Finalnie, otwieramy plik za pomocą edytora tekstowego, aby móc wprowadzić jego zawartość: `touch read_pin.py`, `chmod 777 read_pin.py`, `nano read_pin.py`

Skrypt pokazany jest na **listingu 1**. Po jego skopiowaniu do pliku `read_pin.py` możemy uruchomić skrypt, wpisując w terminalu: `sudo python3 read_pin.py`

Jeśli wszystko pójdzie dobrze skrypt będzie czekał na wykrycie obiektu w barierze optycznej. Gdy laser zostanie zasłonięty, skrypt wydrukuje informacje w terminalu. Aby wyłączyć skrypt należy nacisnąć Ctrl+C.

Strona internetowa do monitorowania obrazu wideo

Teraz musimy skonfigurować interfejs do przeglądania obrazu z kamery internetowej. W tym kroku należy zainstalować serwer WWW Apache. W pierwszej kolejności uruchamiamy instalację poleceniem:

`sudo apt-get install apache2`

Po zainstalowaniu można wpisać w przeglądarce na komputerze *localhost* i wyświetlić stronę powitalną Apache. Jeśli po raz pierwszy konfigurujesz serwer sieciowy, najlepiej jest skonfigurować komputer jednopłytkowy tak, aby miał statyczny adres IP, tj. tak, by router zawsze przypisywał mu ten sam adres IP. Dodatkowo, jeśli chcemy mieć dostęp do systemu z zewnątrz sieci, konieczne jest skonfigurowanie przekierowania portów na routerze tak, by port(y) ze strumieniem wideo były dostępne z zewnątrz sieci lokalnej.

```
Listing 1. Kod skryptu monitorującego stan linii GPIO

laser_1_input = 25 # ta zmienna przechowuje numer pinu, gdzie podłączony jest sensor lasera
import RPi.GPIO as GPIO # biblioteka potrzebna, by mieć dostęp do pinów GPIO
import time # biblioteka z funkcją do usypiania skryptu (sleep)

def laser_a_rising_event(channel):
    # Jest to funkcja, która jest uruchamiana, gdy wiązka lasera
    # zostanie przerwana - na zadanym pinie pojawi się zbrocze opadające
    print("Function CALLED")
    # oczekiwanie mające na sprawdzenie, czy sygnał przerwania jest nadal obecny
    time.sleep(.2)
    if GPIO.input(laser_1_input):
        # jeżeli pin jest nadal w stanie niskim, to na pewno przerwano barierę laserową
        hr = time.strftime("%H", time.localtime()) # godzina
        min = time.strftime("%M", time.localtime()) # minuta
        sec = time.strftime("%S", time.localtime()) # sekunda
        print("Tripped @ ",hr,":",min,":",sec)
    else:
        print("False positive")
    return

print("Setting Up Pins")
# konfiguracja konwencji nazywania pinów
GPIO.setmode(GPIO.BCM)
# ustaw skonfigurowany pin jako wejście
GPIO.setup(laser_1_input,GPIO.IN)
# ustaw wartości domyślne dla detekcji zdarzeń
GPIO.remove_event_detect(laser_1_input)
# dodaj nowe zdarzenie i jego callback
GPIO.add_event_detect(laser_1_input,GPIO.RISING,callback=laser_a_rising_event)
print("Waiting For Event")

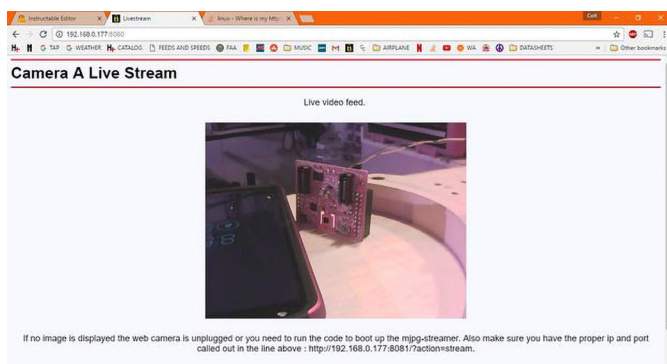
while True:
    time.sleep(.1) # bierne oczekiwanie, by nie zajmować niepotrzebnie procesora

try:
    time.sleep(1) # bierne oczekiwanie, by nie zajmować niepotrzebnie procesora
except KeyboardInterrupt:
    GPIO.cleanup() # zwolnienie zasobów GPIO
    exit()
```

Listing 2. Kod minimalistycznej strony internetowej

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="/mystyle.css">
</head>
<title>Strumień wideo</title>
<body class="home">
<hr class="top"></hr>
<h1 class="black">Strumień wideo</h1>
<hr class="top"></hr>
<div class="content">
<p class="bodytext" align="center">
Podgląd na żywo z kamery w Raspberry Pi.
<br>
</img></a>
Jeśli nie wyświetla się żaden obraz, kamera jest odłączona lub MJPG-
Streamer niewiączony. Upewnij się też, że podano poprawny adres IP i numer portu.
</p>
</div>
</body>
</html>
```

Na **listingu 2** zaprezentowano typową, minimalistyczną stronę internetową, której można użyć do wyświetlania strumienia wideo. Strona korzysta z arkusza stylów CSS *mystyle.css*, który pobrać można ze strony z projektem. Jego brak sprawi, że strona korzystać będzie z domyślnego wyglądu, bez formatowania. Pliki musimy umieścić w folderze strony www, którego adres skonfigurowany jest w pliku konfiguracyjnym Apache. Domyślna lokalizacja to `cd/var/www`. Jeśli chcemy zmienić ich lokalizację, to konieczna jest edycja pliku konfiguracyjnego Apache; wpisujemy w terminalu:



Fotografia 5. Przykładowy widok strony internetowej, prezentującej strumień wideo z kamery podłączonej do Raspberry Pi

Tabela 1. Kluczowe parametry skryptu do wysyłania alertów na e-mail

Nazwa parametru w skrypcie	Znaczenie
<code>your_ip</code>	Adres IP Raspberry Pi
<code>your_ip_optional_port</code>	Port, na którym działa Apache
<code>port_to_camera</code>	Port, gdzie dostępny jest strumień wideo z kamery
<code>sending_email_username</code>	Nazwa użytkownika konta, z którego wysyłany jest e-mail
<code>sending_email_password</code>	Hasło użytkownika konta, z którego wysyłany jest e-mail
<code>to</code>	Adres, na jaki ma być wysłany alert

`sudo nano /etc/apache2/sites-enabled/000-default`

W pliku zmieniamy `/var/www/` na zadaną przez nas lokalizację. Autor zmienił ją na `/home/pi/Desktop/www/`.

Po zapisaniu zmian i skopowaniu plików do zdefiniowanego folderu można przejść do przeglądarki internetowej, gdzie wpiszemy adres IP Raspberry Pi. Naszym oczom powinna ukazać się strona, taka jak na **fotografii 5**. Jeśli chcemy zmodyfikować stronę, na przykład, by wyświetlać więcej strumieni wideo, należy powielić bloki ``, uzupełniając je informacjami takimi jak adres IP i numer portu, gdzie dostępny jest strumień wideo.

E-mailowe alerty

Kolejnym elementem systemu zabezpieczeń, jaki dodamy, jest możliwość wysyłania alertów przez wiadomości e-mail. System będzie wysyłał e-mail na predefiniowany adres w momencie, gdy jakiś obiekt zasłoni laser w barierze. Skrypt napisany w Pythonie, pokazany na **listingu 3**, pozwala realizować to zadanie.

Na potrzeby działania skryptu dobrze jest założyć osobne konto e-mail, gdyż wymaga ono zdjęcia pewnych zabezpieczeń, aby mógł korzystać z serwera SMTP. W skrypcie jest szereg miejsc do edycji i wprowadzenia własnych parametrów. Najważniejsze z nich zapisane są w **tabeli 1**.

Parametr `your_ip_optional_port` najprawdopodobniej będzie pusty, chyba, że twój serwer Apache działa na porcie innym niż domyślny 80. Autor dodał ten parametr, ponieważ na standardowym porcie działa już inny serwer, więc Raspberry Pi oferuje usługę Apache na porcie 8080. Z kolei `port_to_camera` jest portem, na którym strumień wideo nadaje MJPG-Streamer.

Po uzupełnieniu tych zmian na początku skryptu uruchom program, wprowadzając następujące polecenie w terminalu (gdzie `send_mail.py` to nazwa skryptu):

`sudo python3 send_mail.py`

Jeśli wszystko zostało skonfigurowane poprawnie, to skrypt wyśle e-mail na wskazany adres, w treści umieszczając zdjęcie z kamery. Możliwe jest zmodyfikowanie

Listing 3. Skrypt do wysyłania alertów na e-mail

```
#!/usr/bin/python3
import smtplib
import urllib.request
import time
import os
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.image import MIMEImage

# Adres IP Raspberry Pi
your_ip = "192.168.0.177"
# Port ,na którym działa Apache
your_ip_optional_port = ":8080"
# Port, gdzie dostępny jest strumień wideo z kamery
port_to_camera = ":8081"
# Nazwa użytkownika konta, z którego wysyłany jest e-mail
sending_email_username = "your_mail_username"
# Hasło użytkownika konta, z którego wysyłany jest e-mail
sending_email_password = "your_password"
# Adres, na jaki ma być wysłany alert
to = "receiving_email@gmail.com"

try:
    print("create email")
    msg = MIMEMultipart()
    msg['Subject'] = "OMFG Intruder!"
    msg['From'] = "Laser Security System"
    msg['To'] = "OneBadApple"
    msg.preamble = "this is the preamble"

    html = """
    <html>
    <head></head>
    <body>
    <p> Warning! Laser trip wire was triggered.
    <br>The snapshot below was taken at the time of the event.
    <br>To view a live stream of the camera click on the picture or the link.

    <br>
    <br>
    <a href="http://""" + your_ip + your_ip_optional_port + """/index.html">
    
    </a>
    <br>
    <br>
    <a href="http://""" + your_ip + your_ip_optional_port + """/index.html">
    Show me the intruder!
    </a>
    </p>
    </body>
    </html>
    """

    try:
        print("take snapshot")

        # Jeśli skrypt wiesz się na kolejnej linii, to upewnij się,
        # że plik /etc/network/interfaces zawiera następujące linie:
        # auto lo
        # iface lo inet loopback

        file = urllib.request.urlopen("http://" + your_ip + port_to_camera + "?action=snapshot").read()
        print("format image")
        msgImage = MIMEImage(file)
        print("done format image")

    except Exception:
        print("couldnt read image")
        os.system("ifdown eth0")
        time.sleep(2)
        os.system("ifup eth0")
        exit()

    print("build the email")
    msgImage.add_header('Content-ID', '<image1>')
    msg.attach(msgImage)

    html_part = MIMEText(html, "html")
    msg.attach(html_part)

    print("send the email")
    mserv = smtplib.SMTP()
    # zakomentuj tę linię, jeśli chcesz wyczyścić okno terminal
    # to wyjście przydaje się głównie do debugowania
    mserv.set_debuglevel(1)
    mserv.connect('smtp.gmail.com', 587)

    print("successful connection")

    mserv.ehlo()
    mserv.starttls()

    print("log into sending email")
    mserv.login(sending_email_username, sending_email_password)
    print("successful login")

    mserv.sendmail(sending_email_username, to, msg.as_string())
    mserv.close()
    print("sent email")
    os._exit(0)
except Exception as e:
    print(e)
    print("Coult not send mail for laser!")
    print("Check network connection.")
    os._exit(0)
```

Listing 4. Skrypt automatycznie uruchamiający wszystkie podprogramy i kamerę internetową systemu alarmowego

```
#!/bin/bash
cd /home/pi/Desktop/security
sudo python3 read_pin_with_mail.py &
cd /home/pi/Desktop/mjpg-streamer/mjpg-streamer
./mjpg_streamer -i „./input_uvc.so -y /dev/video0” -o „./output_http.so -w ./www -p 8081” &
```

treści e-maila, aby zawierał dodatkowe informacje lub więcej zdjęć z większej liczby kamer (pamiętaj, aby dodać odpowiednio więcej numerów portów, gdzie wyświetlane są strumienie wideo).

Teraz, gdy komputer jest już w stanie wysyłać wiadomości e-mail, należy całość zautomatyzować tak, by za każdym razem, gdy laser był wyłączany, wysyłany był alert e-mail. Należy w tym celu zmodyfikować skrypt `read_pin.py` z listingu 1, dopisując do niego nową funkcję uruchamiania skryptu do wysyłania poczty. Dodatkowa linia wygląda w następujący sposób:

```
sm_pid = os.spawnlp (os.P_NOWAIT, „/usr/bin/
python3”, „python3”, „/home/pi/Desktop/security/
send_mail.py”)
```

Funkcja ta uruchomi skrypt `send_mail.py` równoległe ze skryptem obsługującym barierę laserową. Asynchroniczne uruchamianie tego skryptu jest celowe, ponieważ skrypt do wysyłania wiadomości e-mail pracuje nawet kilka sekund i więcej, i może blokować działanie skryptu obsługi sensora laserowego do momentu zakończenia wysyłania wiadomości e-mail. Nie stanowi to problemu dla systemu z pojedynczą kamerą, ale jeśli w układzie ma być wiele kamer i wiele sensorów, to konieczne jest wykrywanie wszystkich sytuacji z nimi związanych.

Zmienna `sm_pid` będzie zawierać PID procesu (numer identyfikacyjny procesu w systemie) uruchamianego przez to polecenie. Sprawdzając ten PID w dalszej części kodu można sprawdzić, czy wykonywanie skryptu zostało zakończone (i np. mamy do czynienia

z kolejnym zakłóceniem bariery, czy to nadal ta sama sytuacja i nie ma potrzeby uruchamiania skryptu do wysyłania e-maila ponownie).

Skrypt inicjalizujący system po włączeniu komputera

Na tym etapie projekt jest w większości ukończony. Ostatnim krokiem jest uproszczenie uruchamiania systemu za pomocą prostego skryptu, który automatycznie uruchomi wszystkie jego elementy – wszystkie podprogramy i skonfiguruje kamerę internetową. Skrypt bashowy, pokazany na **listingu 4**, można uruchomić podczas uruchamiania systemu, edytując plik `/etc/rc.local` w systemie Raspbian. Musisz zmienić rozszerzenie pliku i być może będziesz musiał zmodyfikować skrypt, aby zawierał odpowiednie ścieżki, jeśli umieścisz skrypty w innym miejscu.

Podsumowanie

Powyższy system tak naprawdę składa się z szeregu niezależnych komponentów, które można zastosować we własnych projektach. Bariery optyczną można wykorzystać w innym systemie zabezpieczeń, podobnie jak skrypt wysyłający e-maile czy strumieniowanie wideo do sieci. Jakkolwiek system ten jest pewną funkcjonalną całością, to może być również źródłem pomysłów do innych konstrukcji bazujących na Raspberry Pi.

Nikodem Czechowski, EP

Źródło: <https://bit.ly/2LHPKL2>

REKLAMA

COMPUTER CONTROLS

Autoryzowany dystrybutor Altium w Polsce



Altium 365

Jedyna na świecie platforma w chmurze, wspomagająca projektowanie i wymianę danych projektów PCB. Bezproblemowo dostępna z Altium Designer lub dowolnego urządzenia podłączonego do internetu.

Bezpłatna dla posiadaczy Subskrypcji Altium.

SUPER CENY
SUPER RABATY

Altium 365 dla posiadaczy Altium Designer **GRATIS**

AD20 + A365

Computer Controls Sp. z o.o.
Bielsko-Biała, ul. Budowlanych 1

tel.: +48 (33) 485 94 90

e-mail: info@ccontrols.pl
www.ccontrols.pl