

# Urządzenia peryferyjne - laboratorium

Karol Kulawiec 241281

Jakub Kalina 241346

10.01.2020

## 1 Wstęp

Celem wykonanego ćwiczenia było zapoznanie się z platformą komputerową noszącą nazwę Raspberry Pi w wersji 4 oraz wykorzystanie jej do stworzenia układów mogących sterować diodą LED oraz innymi elementami zestawu Picoboard.

Raspberry Pi jest to platforma będąca komputerem małych rozmiarów, stworzona między innymi dla robotyków i elektroników, posiadająca dodatkowo piny wejścia i wyjścia umożliwiające sterowanie podłączonymi układami.

Premiera tego projektu miała miejsce 29 lutego 2012 roku, a najnowszą obecnie wersją jest Raspberry Pi 4, z którym mieliśmy możliwość pracować na zajęciach. Wspiera ona różne systemy operacyjne, między innymi Raspbian, Ubuntu Mate, Ubuntu Core, Arch Linux i wiele innych. Wersja używana podczas laboratoriów posiadała zainstalowany system Raspbian, a programy sterujące stworzonymi przez nas układami zostały napisane w języku Python. Dodatkowo jako środowisko programistyczne ułatwiające nam utworzenie programów w języku Python wybraliśmy Thonny IDE.

Każdy z komputerów z gamy Raspberry Pi posiada 40 pinów GPIO (General Purpose Input Output), czyli pinów cyfrowych, a co za tym idzie posiadających dwa stany 1 oraz 0. Mogą one posiadać kierunek odbioru lub wysyłania prądu, możemy kontrolować ich stan oraz kierunek używając wspomnianego wyżej języka Python. Piny GPIO pracują pod napięciem 3,3V wraz z maksymalnym obciążeniem prądowym, które wynosi 16 mA. Wśród nich znajdują się też piny, które zapewniają bezpośredni dostęp do zasilania napięciem 5V pochodzącego z zasilacza sieciowego. Płytkę Raspberry Pi może być zasilana bezpośrednio z tych pinów, a także może zasilać inne urządzenia 5V. W momencie użycia tych wyprowadzeń należy zachować ostrożność z tego powodu, że omijają one wszelkie funkcje bezpieczeństwa, takie jak regulator napięcia, czy bezpiecznik. Piny o oznaczeniu 3V zapewniają stabilne zasilanie napięciem 3.3V. Są one bardzo przydatne przy wstępnym testowaniu niektórych części tworzonych układów oraz przy zasilaniu elementów wymagających stałego zasilania o takim napięciu.

W celu obsługi pinów GPIO w języku Python, konieczne jest zaimportowanie odpowiedniej biblioteki to umożliwiającej. W naszych projektach wykorzystaliśmy bibliotekę RPi.GPIO.

## 2 Opis wykonanych systemów

Podczas zajęć laboratoryjnych udało nam się zaplanować, złożyć i przetestować dwa układy:

1. układ zapalający i gaszący diodę LED w wyznaczonych przez nas odstępach czasowych,
2. układ składający się z ultradźwiękowego czujnika odległości HC-SR04, buzzera oraz diody LED. Jego zadaniem było wytworzenie dźwięku oraz jednoczesne zapalenie diody w momencie, gdy pewien przedmiot zbliżył się do czujnika na określoną przez nas odległość.

## 3 Opis działania programów

- Układ zapalający i gaszący diodę LED

Listing 1: kod programu

---

```

1 import RPi.GPIO as GPIO
2 from time import sleep
3
4 LEDpin = 21
5 GPIO.setmode(GPIO.BCM)
6 GPIO.setup(LEDpin, GPIO.OUT)
7
8 while True:
9     GPIO.output(LEDpin, GPIO.HIGH)
10    sleep(1)
11    GPIO.output(LEDpin, GPIO.LOW)
12    sleep(1)

```

---

Na początku wczytujemy bibliotekę RPi.GPIO, aby móc odwoływać się do konkretnych pinów na płycie oraz sleep z biblioteki time, aby opóźnić zmianę stanu diody, w celu uwidocznienia rezultatów. Następnie pod zmienną LEDpin ustawiamy nr pinu, do którego podpieliśmy diodę. W linii 5 ustawiamy GPIO.setmode na GPIO.BCM, oznacza to, że będziemy się odnosić do portów na płycie wg numerów, występujących po GPIO. Następnie ustawiamy ten port jako wyjście.

Główna część programu polega na zapętleniu kolejno ustawiania tego portu na stan wysoki oraz na stan niski.

- Układ informujący o zbliżeniu się do czujnika na określoną odległość

Listing 2: kod programu

---

```

1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM)
5 GPIO.setwarnings(False)
6 TRIG = 5
7 ECHO = 21
8 BUZZ = 23
9 GPIO.setup(BUZZ, GPIO.OUT)
10 GPIO.setup(TRIG, GPIO.OUT)
11 GPIO.setup(ECHO, GPIO.IN)
12 while True:
13     GPIO.output(TRIG, True)
14     time.sleep(0.0001)
15     GPIO.output(TRIG, False)
16
17     while GPIO.input(ECHO) == False:
18         start = time.time()
19
20     while GPIO.input(ECHO) == True:
21         end = time.time()
22
23     sig_time = end - start;
24
25     #cm:
26     distance = sig_time / 0.000058
27
28     if (distance < 10):
29         GPIO.output(BUZZ, True)
30     else:
31         GPIO.output(BUZZ, False)
32
33     print('Distance: {} cm'.format(distance))
34     time.sleep(0.1)
35

```

---

Tak jak w poprzednim programie ustawiamy wartości początkowe, dodatkowo setwarnings na False, aby wyłączyć ostrzeżenia. Czujnik HC-SR04 posiada 4 piny sygnałowe. Dwa z nich służą do zasilania układu (Vcc, GND), a drugie dwa (trigger i echo) do wykonywania pomiarów. Trigger, to wejście wyzwalające. Gdy podamy na nie stan wysoki (przez min. 10 mikrosekund), to rozpocznie się pomiar odległości. Natomiast z wyjścia echo odczytamy zmierzoną odległość. Maksymalny zasięg tego układu deklarowany jest przez producenta na 4m. Dodatkowo w programie zastosowaliśmy Buzzer, którego zasilamy w momencie, gdy zmierzony dystans będzie mniejszy niż 10 cm oraz diodę LED, zasilaną w tym samym czasie, co Buzzer. Główna pętla programu polega na ustawieniu na 100  $\mu$ s wejścia TRIG na stan wysoki, a następnie na stan niski, aby rozpocząć pomiar. Następnie ustawiamy początek pomiaru czasu na ostatni moment, gdy pin ECHO ma jeszcze stan niski, a koniec pomiaru czasu na ostatni moment, gdy pin ECHO ma stan wysoki. Czas przebywania pinu ECHO w stanie wysokim jest proporcjonalna do odległości. Obliczamy go na podstawie różnicy w czasie końca pomiaru i początku pomiaru. Uzyskany wynik dzielimy przez 0.000058, aby uzyskać odległość w centymetrach.

Gdy odległość jest mniejsza niż 10cm, Buzzer oraz dioda LED dostają zasilanie. Ostatecznie jeszcze wypisujemy na ekranie odległość, a na końcu zatrzymujemy program na 0.1 sekundy, aby ograniczyć ilość wykonywanych pomiarów w czasie.

## 4 Wyniki

Wynik konsolowy:

```
root@raspberrypi:/home/lab229/Desktop# python led.py
Distance: 23.4636767157 cm
Distance: 22.4853384084 cm
Distance: 15.3985516778 cm
Distance: 12.3607701269 cm
Distance: 7.79382113753 cm
Distance: 8.91192206021 cm
Distance: 13.6391869907 cm
Distance: 10.3259908742 cm
Distance: 9.6189564672 cm
Distance: 8.18844499259 cm
Distance: 28.1703883204 cm
Distance: 5.68916057718 cm
Distance: 5.17532743257 cm
Distance: 2379.11734088 cm
Distance: 2376.20698995 cm
Distance: 2374.37774395 cm
Distance: 2373.07055243 cm
Distance: 2374.03655874 cm
Distance: 16.0192621165 cm
Distance: 14.9299358499 cm
Distance: 63.2919114212 cm
```

Rysunek 1: Konsola programu

Link do filmu prezentującego działanie układu: [UP - Raspberry Pi](#)

## 5 Wnioski

Urządzenie Raspberry Pi to świetne urządzenie, posiadające podobne możliwości co komputer, ale w bardziej ograniczonych zasobach. Można stworzyć na nim wiele ciekawych projektów. Podczas zajęć udało nam się stworzyć mini projekt, który korzysta z bazowych funkcjonalności płytki. Praca na nim dzięki bogatej i aktualnej dokumentacji nie stanowi problemu, a dodatkowo sprawia wiele radości.

Cały kod zawarty jest w sprawozdaniu, dlatego brak linku do repozytorium.