

SPRAWOZDANIE

Ćwiczenie 17

RaspberryPi

WYKONANIE:

- Szymon Hutnik 252736
- Damian Mucha 252708

TERMIN ZAJĘĆ: środa TP, 11.15

DATA: 10.11.2021

PROWADZĄCY: dr inż. Tomasz Walkowiak

1. Treść zadania

Celem ćwiczenia było zapoznanie się z budową oraz RaspberryPi oraz płytki Picoboard.

Zadaniem do zrealizowania było zapoznanie się z komendami programu gpio oraz napisanie programu w Pythonie, który pozwalałby na sterowanie układem na płycie. Należało zbudować prosty układ z dostępnych elementów np. diód.

2. Opis urządzenia

RaspberryPi – platforma komputerowa stworzona przez Raspberry Pi Foundation, klasyfikowana jako SBC (ang. *single-board computer*). Obecnie Raspberry czwartej generacji posiada wersję wyposażoną w 8GB pamięci RAM oraz 4 rdzeniowy, 64-bitowy procesor. Tworząc układ użytkownik może dopasować docelową funkcjonalność pod swoje potrzeby, a sama platforma zapewnia ogromne możliwości do eksperymentowania. Pomaga w tym także obsługa licznych portów wejścia oraz wyjścia, czego efektem jest bogata gama możliwych do wykorzystania komponentów. Warto również zauważyć, iż łatwa wymiana kart SD pozwala na szybką i prostą zmianę funkcjonalności urządzenia. Jako mały komputer posiada on również mały pobór mocy (zaleta zarówno ekonomiczna, jak i ekologiczna), a także - niską cenę. Jest to więc świetna opcja do hobbystycznych, niskobudżetowych projektów. Warto wspomnieć, iż Raspberry nie posiada zabezpieczeń przed niepoprawną konstrukcją układu, więc w przypadku niepoprawnego podłączenia pinów możliwe jest zniszczenie platformy, bądź innych komponentów.

Zadaniem realizowanym przez nas jest podłączenie do płytki kilku urządzeń: diod, brzęczka i przycisku i zaprogramowanie układu w taki sposób, żeby diody i brzęczek uruchamiały się cyklicznie co jakiś czas lub w zależności od tego czy przycisk jest włączony.

Aby osiągnąć taki efekt wykorzystane zostaną piny GPIO (General Purpose Input Output), do których zostaną podłączone wspomniane elementy. Mogą one posiadać dwa stany: ON i OFF (jedynek i zero logiczne). Zgodnie z ogólną zasadą konstruowania układów podczas pracy z Raspberry PI, diodę podłączamy poprzez rezystor do dowolnego pinu GPIO (podczas zajęć wykorzystywaliśmy GPIO21), zaś drugą jej nóżkę łączymy z masą.

Obsługa diody w shellu została oparta na 3 komendach:

➤ `echo 21 > /sys/class/gpio/export`

echo 21 (bądź inny numer diody) wpisany do pliku export, która pozwala nam na dostęp do podanego pinu. W przypadku, gdy chcemy wyłączyć dostęp należy wykorzystać plik "przeciwny" - unexport.

➤ `echo out > /sys/class/gpio/gpio21/direction`

echo in/out wpisana do pliku direction (znajdującym się w powstałym dzięki poprzedniej komendzie katalogu gpio21) - ustawia pin jako wejściowy/wyjściowy. W naszym przypadku chcemy sterować diodę, więc użyjemy komendy echo out.

➤ `echo 1 > /sys/class/gpio/gpio21/value`

`echo 1/0` wpisana do pliku `value` - komenda odpowiedzialna za ustawienie stanu wysokiego, bądź niskiego na danym pinie.

Wykorzystując język Python oraz znajdujące się na Raspberry środowisko Thonny Python IDE możliwe jest napisanie bardziej skomplikowanych skryptów. Korzystać będziemy z biblioteki `RPi.GPIO` (pozwalającą na kontrolowanie pinów GPIO).

W tej bibliotece, metodami odpowiadającymi powyższym komendom (2 i 3) są odpowiednio:

- `GPIO.setup(21, GPIO.OUT)`
- `GPIO.output(21, GPIO.HIGH)`

Problematyka implementacji pozostałej części skryptu opiera się wyłącznie na zaprojektowaniu odpowiedniego przepływu sterowania oraz zapisaniu go w języku Python.

3. Program sterujący

```
import RPi.GPIO as GPIO
import time
import sys
import os

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(23, GPIO.OUT)
GPIO.setup(21, GPIO.OUT)
GPIO.setup(26, GPIO.IN)
GPIO.output(21, GPIO.LOW)

for q in range(5):
    print("ON")
    GPIO.output(23, GPIO.HIGH)

    time.sleep(4)

    print("OFF")
    GPIO.output(23, GPIO.LOW)

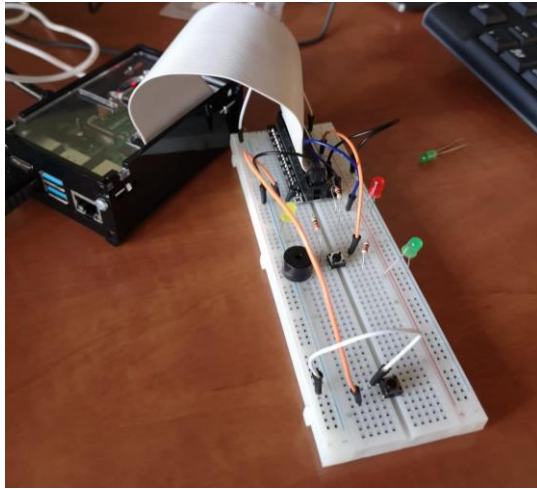
    time.sleep(1)
    if(GPIO.input(26) == 1):
        GPIO.output(21, GPIO.HIGH)

    else:
        GPIO.output(21, GPIO.LOW)

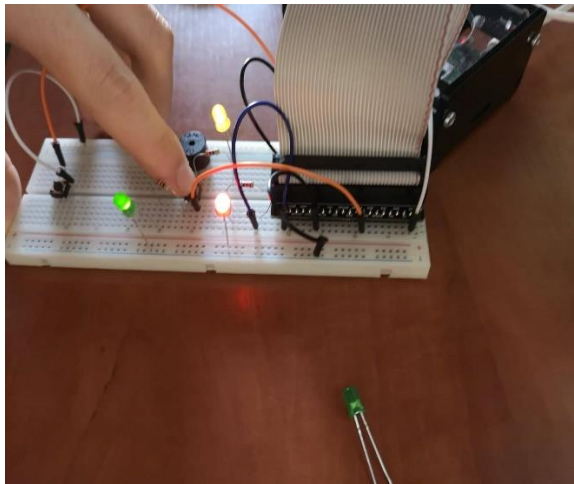
GPIO.cleanup
```

Listing 1: Program w Pythonie

4. Przedstawienie układu



Rysunek 1: Podłączony układ



Rysunek 2: Działający układ

5. Liteatura

- Damian Szymański, *Kurs Raspberry Pi forbot.pl*, 2021
- *Raspberry Pi 4 Model B datasheet*, czerwiec 2019
- Ben Croston, *RPi.GPIO Python Module documentation*, 2021