Wykorzystanie linii GPIO Raspberry Pi 3

Angelika Tefelska Dariusz Tefelski

2023-05-07

Celem ćwiczenia jest zapoznanie z możliwościami programistycznymi obsługi wejścia-wyjścia ogólnego przeznaczenia (GPIO) Rasbpberry PI.

Uwagi wstępne:

- W celu minimalizacji zagrożenia uszkodzenia Raspberry Pi, wszelkie podłączenia należy wykonywać po wyłączeniu zasilania.
- Potrzebne biblioteki należy pobrać korzystając z narzędzia: *apt* na przykład:

```
sudo apt install wiringpi
```

- Jednakże obecnie biblioteka wiringpi może nie być dostępna w repozytorium pakietów systemu, gdyż autor przestał ją już wspierać. Biblioteka jest jednak wspierana przez społeczność w repozytorium github. Wprowadzono w niej także poprawki umożliwiające uruchomienie jej w 64-bit'owym systemie i z nowymi układami SoC: Raspberry Pi 4, 400 itd. Aby pobrać i skompilować bibliotekę należy:
- Zainstalować git

```
sudo apt install git
mkdir ~/repos
cd repos
git clone https://github.com/WiringPi/WiringPi
cd WiringPi
sudo ./build
```

- Moduł do obsługi 1-wire może powodować problemy z kontrolą pinu *GPIO4* (BCM4), ponieważ domyślnie ten pin jest wykorzystywany do transmisji 1-wire. Aby uniknąć problemów należy wstępnie wyładować moduł jądra poleceniem:

```
sudo rmmod w1_gpio
```

Możliwe jest także przemapowanie pinu używanego przez wpisanie do pliku: */boot/config.txt*:

```
dtoverlay=w1-gpio,gpiopin=x
```

gdzie x - oznacza numer pinu, który ma służyć jako 1-wire.

WiringPi

Biblioteka, która ułatwia programowanie portów GPIO Raspberry PI.

Polecenie gpio:

Opis parametrów dostępny na stronie http://wiringpi.com/the-gpio-utility/ Można zapoznać się także wydając polecenie *man gpio*, a właściwie: *man /man/man1/gpio.1* (wygląda na to, że manual został umieszczony w niewłaściwym katalogu.)

```
gpio readall
gpio -g mode 17 out
```

Opcja **-g** oznacza, że posługujemy się numeracją pinów BCM_GPIO, tzn. odpowiadające wyprowadzeniom układu SoC.

Zadanie dotyczące biblioteki wiringpi

Na wykładzie przedstawiono podstawy stosowania biblioteki *wiringPi*. Wykorzystując przykład skryptu bash z wykładu dotyczący obsługi diody LED oraz przycisku, przygotować skrypt, który będzie cyklicznie sterował światłami ulicznymi.

Światła uliczne mają następujące stany:

1. Światło czerwone - zabrania wjazdu za sygnalizator (czas trwania 10s)
2. Światło czerwone + żółte - zabrania przejazdu za sygnalizator, informuje o zmianie na zielone, (czas trwania 3s)
3. Światło zielone - zezwala na wjazd za sygnalizator (czas trwania 10s)
4. Światło żółte - zabrania wjazdu za sygnalizator, informuje o zmianie na czerwone (czas trwania 3s)

Wykorzystać:

- piny: GPIO16 (36), GPIO20 (38), GPIO21 (40)
- 3 x rezystory 330 Ω
- Diody LED: czerwona, żółta, zielona

Wykorzystanie modulacji PWM

Do sprawdzenia działania modulacji PWM możemy wykorzystać serwo modelarskie. Serwo wymaga aby częstotliwość fali wynosiła 50 Hz (okres 20 ms). Natomiast czas występowania stanu wysokiego mieści się najczęściej w zakresie od 600 μs do 2400 μs. Serwo należy podłączyć do Raspberry PI następująco:

Wyprowadzenie Serwomechanizmu	Pin GPIO Raspberry PI (nr pinu na złączu)
Przewód czarny (GND)	GND (pin 6)
Przewód czerwony (+5V)	+5V (pin 2)
Przewód biały (PWM)	GPIO24 (pin nr 18)

Napisać program *servo.c*:

```
#include <wiringPi.h>
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <softPwm.h>

int main(int argc, char* argv[]){
  int pos;
  if(argc != 2){
    printf("Usage: servo <value 5-20>");
    exit(1);
  }
  pos = atoi(argv[1]);
    if ( wiringPiSetupGpio() == -1 ){
        printf("Cannot get GPIO!");
        exit(1);
    }

    pinMode(24, OUTPUT);
    digitalWrite(24, LOW);
    softPwmCreate(24,0,200);
```

```
    softPwmWrite(24,pos);
    delay(1000);
    return 0;
}
```

Do kompilacji warto przygotować plik *Makefile* o zawartości:

```
all:
      gcc -o servo servo.c -lwiringPi
```

Wykonać budowanie:

```
make
```

i uruchomić badając wartości w zakresie <5 - 20>.

```
./servo 10
```

Biblioteka pigpio

Biblioteka wymaga uruchomionego daemona - pigpiod, ale jest lepiej przystosowana np. do obsługi software-owego PWM.

Przed wykonaniem programu, musi być uruchomiony daemon przez polecenie:

```
sudo pigpiod
```

Ewentualnie można umieścić to wywołanie w skrypcie startowym np. /etc/rc.local aby uruchamiał się automatycznie w trakcie uruchamiania systemu.

Komenda pigs

W bibliotece pigpio istnieje także polecenie, którym można kontrolować GPIO Raspberry Pi z linii poleceń - *pigs*.

Sprawdzenie działania software-owego PWM można wykonać poleceniem:

```
pigs s 24 1000
```

Gdzie wartości dla serwomechanizmu zamiast 1000 (μs), można wpisać w zakresie od około 600 do 2400 μs.

Zadanie - Gra Reakcyjna (Python i biblioteka gpiozero)

Projekt umożliwia zbudowanie prostej reakcyjnej gry dla 2 graczy. Gra zawiera pułapki, podlicza wynik i wskazuje zwycięzcę.

Potrzebne:

- Raspberry Pi
- Płytka prototypowa
- 2 x dioda LED czerwona
- 2 x przycisk
- 1 x dioda LED RGB
- 5 x 330Ω rezystory
- 12 x przewód do połączeń wewnętrznych na płytce prototypowej

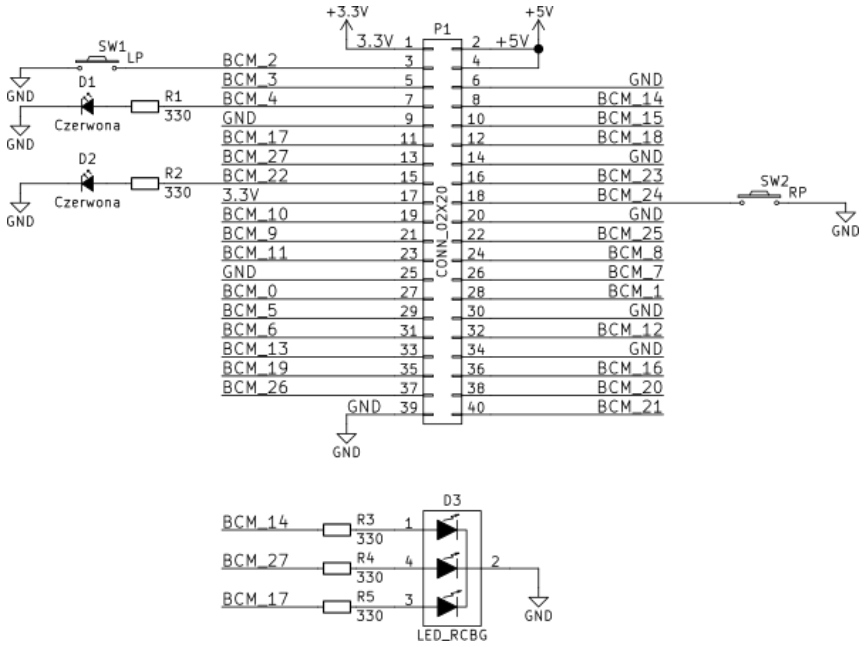

Figure 1: Schemat połączeń gry reakcyjnej

Instrukcja wykonania

1. Upewnij się, że Raspberry PI jest wyłączone z zasilania, wykonaj obwód na płytce prototypowej i podłącz do Raspberry PI.
2. Włącz Raspberry Pi
3. Wyładuj kolidujący moduł jądra - patrz *uwagi* na początku instrukcji i uruchom ponownie Raspberry Pi
4. Włącz Raspberry Pi i pobierz przykładowy kod programu

```
git clone https://github.com/sq5nbg/rpesk-advanced
sudo apt-get update
sudo apt-get install python3-gpiozero python3-colorzero python3-pkg-resources
cd rpesk-advanced
sudo python 02_reaction_game.py
```

5. Gra przeznaczona jest dla 2 graczy. Polega na jak najszybszym wciśnięciu przycisku, jeśli dioda LED zaświeci się innym kolorem niż czerwony. Wciśnięcie przycisku gdy pojawi się kolor czerwony powoduje odjęcie jednego punktu od liczby punktów gracza. Diody na płytce wskazują który gracz wygrywa. Jeśli świecą się 2 jest remis.

Gra typu papier, nożyczki, kamień

Zaprojektuj podobną grę - ale dla jednego gracza i wykonaj z wykorzystaniem języka Python oraz biblioteki gpiozero oraz dostępnych elementów.

Gra ma być oparta na zasadzie “Papier - Nożyczki - Kamień” i ma rozgrywać się pomiędzy graczem a Raspberry PI.

Wykorzystać co najmniej:

- 3 przyciski
- 2 diody LED + 2 rezystory 330Ω

Generalna zasada polega na tym, że użytkownik wybiera - za pomocą przycisku: papier, nożyczki albo kamień.

Raspberry Pi losuje swój wybór.

Mamy następujące zależności:

1. Jeśli użytkownik wybrał papier:
   - Jeśli Raspberry PI wybrało papier - to mamy remis.
   - Jeśli Raspberry PI wybrało kamień - to wygrywa gracz.
   - Jeśli Raspberry PI wybrało nożyczki - to gracz przegrywa.
2. Jeśli użytkownik wybrał nożyczki:
   - Jeśli Raspberry PI wybrało papier - to wygrywa gracz.
   - Jeśli Raspberry PI wybrało kamień - to gracz przegrywa.
   - Jeśli Raspberry PI wybrało nożyczki - to mamy remis.
3. Jeśli użytkownik wybrał kamień:
   - Jeśli Raspberry PI wybrało papier - to gracz przegrywa.
   - Jeśli Raspberry PI wybrało kamień - to mamy remis.
   - Jeśli Raspberry PI wybrało nożyczki - to wygrywa gracz.

Stanowisko laboratoryjne

Podzespoły potrzebne do realizacji ćwiczenia:

- Raspberry Pi
- Serwomechanizm (mikroserwo)
- Płytka prototypowa
- diody LED czerwone
- diody LED żółte
- diody LED zielone
- przyciski
- 1 x dioda LED RGB
- Rezystory 330Ω
- przewody do połączeń wewnętrznych na płytce prototypowej

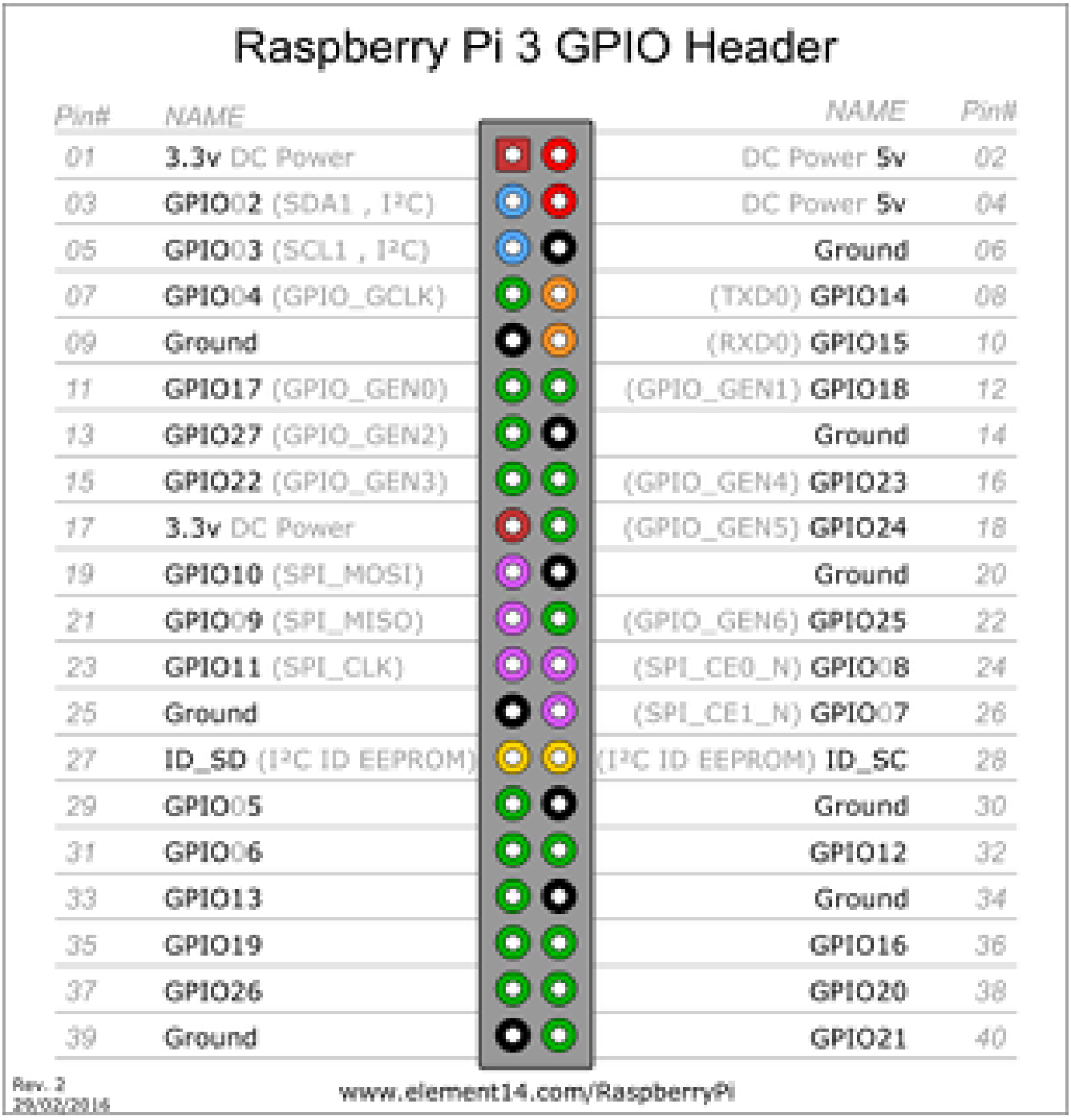

Figure 2: Złącze 40-pin Raspberry Pi.