

Systemy komputerowe w sterowaniu i pomiarach (SKPS)

Instrukcja do laboratorium 2

dr hab. inż. Wojciech Zabołotny, dr inż. Dawid Seredyński

[Informacje wstępne](#)

[Cel laboratorium](#)

[Zadanie domowe](#)

[Zakres wejściówki](#)

[Zadania](#)

[Praca na zajęciach](#)

[Oznaczenia użyte w tym dokumencie](#)

[Dlaczego OpenWRT?](#)

[Problemy, które mogą wystąpić](#)

[Problemy z DNS](#)

[Logowanie za pomocą ssh do systemu ratunkowego na RPi](#)

[Zapoznanie się z systemem OpenWRT](#)

[Skąd pobrać OpenWRT?](#)

[Jak zainstalować OpenWRT?](#)

[Instalacja OpenWRT z wykorzystaniem systemu ratunkowego](#)

[Interfejs HTTP dla OpenWRT](#)

[Eksperymenty w OpenWRT](#)

[Obsługa akcesoriów przez GPIO](#)

[Warto wiedzieć](#)

[Schematy połączeń](#)

[Obsługa GPIO przez sysfs](#)

[Obsługa GPIO w Pythonie](#)

[Stanowisko pracy](#)

[Zadania - obsługa akcesoriów](#)

[Akcesoria](#)

[Literatura](#)

Informacje wstępne

Cel laboratorium

Celem laboratorium jest zapoznanie się ze środowiskiem OpenWRT, sposobem obsługi GPIO za pomocą sysfs, oraz prostymi układami wejścia/wyjścia.

Laboratorium 2 jest za 8 punktów, w tym:

- 2 - zadanie domowe
- 1 - wejściówka (UWAGA: osoby, które otrzymają 0 punktów z wejściówki mogą nie zostać dopuszczone do dalszej części laboratorium)
- 5 - praca na zajęciach

Zadanie domowe

Zadanie domowe należy wykonać przed zajęciami.

Należy zaimplementować w języku Python dwa programy generujące sygnał PWM:

1. o zmiennej częstotliwości od 30Hz do 1000Hz i z powrotem do 30Hz (zmiana może być liniowa lub w formie innej funkcji ciągłej, np. sinus)
2. o zmiennym wypełnieniu od 10% do 90% i z powrotem do 10% (zmiana może być liniowa lub w formie innej funkcji ciągłej, np. sinus)

Wygenerowany sygnał należy zwizualizować w formie wykresu w dziedzinie czasu (np. za pomocą matplotlib), a wizualizację należy zapisać w pliku .png.

Kody źródłowe oraz pliki .png z wizualizacją należy umieścić w repozytorium na gitlab (w tym, które zostało utworzone na pierwszym laboratorium).

UWAGA: Zadanie domowe należy umieścić na repozytorium PRZED rozpoczęciem laboratorium 2.

Zakres wejściówki

1. qemu
2. OpenWRT
3. sysfs
4. polecenia: cat, cp, dd, echo, gzip, ifconfig, losetup, ls, mkdir, mount, wget
5. PWM

Zadania

Plan laboratorium 2:

1. Uruchomienie OpenWRT na RPI 4B
2. Podłączenie podstawowych akcesoriów i ich obsługa przez sysfs oraz za pomocą Pythona - szczegółowy opis jest w sekcji [Zadania - obsługa akcesoriów](#)

Praca na zajęciach

W trakcie zajęć proszę na bieżąco tworzyć raport z wykonanych prac. Kolejne etapy należy przedstawiać prowadzącemu.

Oznaczenia użyte w tym dokumencie

Polecenia, które są wpisywane w konsoli **Linuxa** na komputerze **host (PC)** oznaczone są jasnoszarym tłem, np.:

```
ls -la /
```

Polecenia, które są wpisywane w konsoli **Linuxa** na komputerze **RPI** oznaczone są malinowym tłem (raspberry, czyli malina), np.:

```
ls -la /
```

Dlaczego OpenWRT?

Środowisko Buildroot stanowi bardzo dobre rozwiązanie, jeśli chcemy zbudować system idealnie dostosowany do naszych potrzeb. Ceną jaką za to płacimy, jest jednak konieczność dość czasochłonnej pełnej rekompilacji po wielu zmianach konfiguracji (po niektórych wystarcza częściowa rekompilacja). W warunkach czasowych ograniczeń naszego laboratorium, przydatne może być środowisko OpenWRT, które oferuje możliwość instalowania i odinstalowywania pakietów. Pozwala także skompilować i dodać własny pakiet.

Problemy, które mogą wystąpić

Problemy z DNS

W sieci wewnętrznej Politechniki Warszawskiej zdarzają się problemy DNS. W szczególności występują problemy z otwieraniem strony downloads.openwrt.org oraz z instalowaniem pakietów w OpenWRT.

Problem ten można obejść, definiując lokalnie adres serwera nazw w pliku `/etc/resolv.conf` w systemie ratunkowym (na RPi) oraz w OpenWRT (kiedy już go uruchomimy na RPi).

W pliku

`/etc/resolv.conf`

należy dodać linię:

```
nameserver 8.8.8.8
```

tak, aby był to ostatni zdefiniowany nameserver w tym pliku. Najlepiej dodać tę linię na samym końcu pliku `/etc/resolv.conf`

Dzięki temu, jako pierwszy zostanie zawsze odpytany DNS w sieci PW, a jeśli wystąpi timeout, zapytanie zostanie przekierowane do serwera 8.8.8.8.

W ten sposób będzie można ściągnąć OpenWRT z poziomu systemu ratunkowego poleceniem np. `wget`.

UWAGA: Plik `/etc/resolv.conf` można edytować tylko na systemie ratunkowym (na RPi) oraz w uruchomionym OpenWRT (na RPi), gdyż użytkownicy mają tam uprawnienia `sudo`. Studenci nie mają uprawnień do edycji tego pliku na komputerach PC w laboratorium.

UWAGA: Plik `/etc/resolv.conf` zawiera konfigurację uzyskaną z sieci, więc co jakiś czas może nastąpić jego aktualizacja i nadpisanie wprowadzonych zmian. Jeśli wystąpią

problemy ze ściągnięciem pakietów, to należy sprawdzić, czy dalej jest tam zdefiniowany nameserver 8.8.8.8

Logowanie za pomocą ssh do systemu ratunkowego na RPi

W systemie ratunkowym serwer ssh jest włączony, ale mogą występować problemy z logowaniem ze względu na konfigurację konta użytkownika root na RPi. Jeśli wystąpią problemy z logowaniem, np. pojawi się komunikat "Permission denied, please try again.", to można spróbować zmienić hasło na systemie ratunkowym poleceniem

```
passwd
```

Zapoznanie się z systemem OpenWRT

Skąd pobrać OpenWRT?

Najnowszą aktualnie wersję dla Raspberry Pi 4 możemy znaleźć pod adresem: <https://downloads.openwrt.org/releases/21.02.1/targets/bcm27xx/bcm2711/>

Obraz systemu do instalacji jest pod adresem:

<https://downloads.openwrt.org/releases/21.02.1/targets/bcm27xx/bcm2711/openwrt-21.02.1-bcm27xx-bcm2711-rpi-4-ext4-factory.img.gz>

UWAGA: proszę korzystać z OpenWRT w wersji 21.02.1. W nowszej wersji zaobserwowaliśmy problemy z obsługą GPIO przez sysfs.

Jak zainstalować OpenWRT?

Gdybyśmy mieli wolną kartę SD i gdybyśmy mieli PC z uprawnieniami roota, to moglibyśmy po prostu wgrać obraz na kartę (na przykład rozpakowując go w locie)

```
cat owrt.img.gz | gzip -c -d | dd of=/dev/sdX
```

Podobnie jak w przypadku obrazu karty tworzonego przez Buildroot, taka operacja spowoduje utworzenie na karcie SD dwóch partycji - boot z systemem VFAT i rootfs z systemem ext2. Możemy oczywiście rozszerzyć partycję ext2 poleceniami fdisk i resize2fs. W laboratorium mamy jednak jedną kartę zawierającą system ratunkowy. Czy możemy zainstalować OpenWRT nie tracąc systemu ratunkowego?

Instalacja OpenWRT z wykorzystaniem systemu ratunkowego

Pobieramy obraz systemu poleceniem wget (w jednej linii!):

```
wget
```

```
https://downloads.openwrt.org/releases/21.02.1/targets/bcm27xx/bcm2711/openwrt-21.02.1-bcm27xx-bcm2711-rpi-4-ext4-factory.img.gz
```

Rozkompresowujemy go:

```
gzip -d openwrt-21.02.1-bcm27xx-bcm2711-rpi-4-ext4-factory.img.gz
```

Ładujemy jako urządzenie "loop":

```
losetup -P -f openwrt-21.02.1-bcm27xx-bcm2711-rpi-4-ext4-factory.img
```

Urządzenie "loop" to urządzenie blokowe, które reprezentuje bloki danych z pliku lub urządzenia blokowego.

Należy wyświetlić stan wszystkich urządzeń loop, poleceniem:

```
losetup -a
```

i odnaleźć nazwę urządzenia loop, na którym został wczytany obraz systemu.

Poniżej założono, że partycje obrazu zostały zamontowane przez urządzenie loop0 (za pomocą polecenia losetup).

Kopiujemy partycję z rootfs z obrazu OpenWRT na kartę SD, na partycję 2:

```
dd if=/dev/loop0p2 of=/dev/mmcblk0p2 bs=4096
```

te dwie ostatnie litery mówi o partycjach konkretnych, w tym przypadku s to drugie partycje

Tworzymy katalogi, w których zamontujemy partycje:

```
mkdir /mnt/boot /mnt/owrt
```

Montujemy partycję pierwszą obrazu OpenWRT w katalogu /mnt/owrt:

```
mount /dev/loop0p1 /mnt/owrt
```

Montujemy partycję pierwszą karty SD (partycja boot) w katalogu /mnt/boot:

```
mount /dev/mmcblk0p1 /mnt/boot
```

Kopiujemy pliki z obrazu OpenWRT do katalogu user na partycji boot na karcie SD:

```
cp /mnt/owrt/cmdline.txt /mnt/boot/user/
```

```
cp /mnt/owrt/kernel8.img /mnt/boot/user/
```

```
cp /mnt/owrt/bcm2711-rpi-4-b.dtb /mnt/boot/user/
```

Możemy teraz powiększyć system plików OpenWRT tak, żeby wypełnił całą partycję:

```
resize2fs /dev/mmcblk0p2
```

Teraz możemy już zrestartować system poleceniem

```
reboot
```

UWAGA: aby uruchomić OpenWRT, którego obraz właśnie został wgrany do katalogu /user na partycji boot (mmcblk0p1), należy podczas bootowania przytrzymać przycisk SW4 na adapterze do RPi.

Powinien się zgłosić nam system OpenWRT. Musimy w nim skorygować konfigurację sieci (niestety MUSIMY użyć edytora vi, bo sieć jeszcze nie działa, więc nie mamy jak zainstalować innego edytora):

```
vi /etc/config/network
```

Zawartość tego pliku będzie wyglądać mniej więcej tak:

```

config interface 'loopback'
    option device 'lo'
    option proto 'static'
    option ipaddr '127.0.0.1'
    option netmask '255.0.0.0'

config globals 'globals'
    option ula_prefix 'fd28:3901:e94c::/48'

config device
    option name 'br-lan'
    option type 'bridge'
    list ports 'eth0'

config interface 'lan'
    option device 'br-lan'
    option proto 'static'
    option ipaddr '192.168.1.1'
    option netmask '255.255.255.0'
    option ip6assign '60'

```

Usuwamy, lub komentujemy sekcję device i zmieniamy konfigurację interfejsu “lan”:

```

config interface 'loopback'
    option device 'lo'
    option proto 'static'
    option ipaddr '127.0.0.1'
    option netmask '255.0.0.0'

config globals 'globals'
    option ula_prefix 'fd28:3901:e94c::/48'

#config device
#    option name 'br-lan'
#    option type 'bridge'
#    list ports 'eth0'

config interface 'lan'
    option device 'eth0'
    option proto 'dhcp'
#    option ipaddr '192.168.1.1'
#    option netmask '255.255.255.0'
#    option ip6assign '60'

```

Zapisujemy zmiany, a następnie restartujemy sieć poleceniem:

```
/etc/init.d/network reload
```

System OpenWRT jest gotów do pracy.

Interfejs HTTP dla OpenWRT

Konfigurację oraz podgląd stanu uruchomionej maszyny z OpenWRT, a także zdalną instalację pakietów można przeprowadzić za pomocą interfejsu HTTP. W tym celu należy w przeglądarce, na komputerze PC wpisać adres:

`http://<ip_RPi>`

login: root

hasło: (puste)

UWAGA: RPi musi być uruchomione z systemem OpenWRT, oraz połączenie sieciowe między RPi oraz PC musi być poprawnie skonfigurowane.

Eksperymenty w OpenWRT

Jakie proste eksperymenty możemy zrobić w OpenWRT?

`opkg install nazwa_pakietu`
`opkg list-installed`

Polecenie

`opkg update`

aktualizuje listę pakietów (można je także wykonać zdalnie przez [Interfejs HTTP dla OpenWRT](#)). Jeśli chcemy komunikować się z układami podłączonymi przez GPIO, może się nam przydać pakiet `gpiod-tools`. Odpowiednio, dla układów podłączonych przez interfejsy SPI i I2C, warto zainstalować `i2c-tools` i `spi-tools`. Zawartość pakietu możemy sprawdzić poleceniem

`opkg files <nazwa pakietu>`

Wyszukiwanie pakietów zapewniają polecenia `opkg find` i `opkg search`. W OpenWRT jednak, do znalezienia pakietu, zawierającego słowo1 i słowo2 w opisie może lepiej przydać się polecenie:

`opkg list | grep słowo1 | grep słowo2`

Do obsługi urządzeń I/O z poziomu Pythona, mogą się przydać pakiety `python3-smbus`, `python3-gpiod` oraz `gpio4` (instalowany przez `pip`).

Obsługa akcesoriów przez GPIO

W tej części laboratorium należy zapoznać się z obsługą GPIO przez `sysfs` oraz z poziomu programu napisanego w języku Python. Nadal pracujemy w OpenWRT.

Warto wiedzieć

Schematy połączeń

Należy korzystać z oprogramowania do tworzenia schematów połączeń, np. Fritzing (<https://fritzing.org>).

Model płytki RPi 4B dla Fritzing jest dostępny na Moodle (źródło: <https://forum.fritzing.org/t/raspberry-pi-4-model-b/8622/26>).

UWAGA:

1. Przed połączeniem akcesoriów do RPi należy obowiązkowo stworzyć schemat.
2. Po zatwierdzeniu przez prowadzącego można wykonać połączenia na sprzęcie (koniecznie przy wyłączonym zasilaniu!).
3. Przed włączeniem zasilania, należy uzyskać zgodę prowadzącego.

Należy znać i rozumieć zasady podłączania LED, buzzera, przycisku i innych akcesoriów. Należy wiedzieć co się robi i dlaczego.

Ogólna zasada jest taka, że należy sobie zadać pytania i na nie odpowiedzieć:

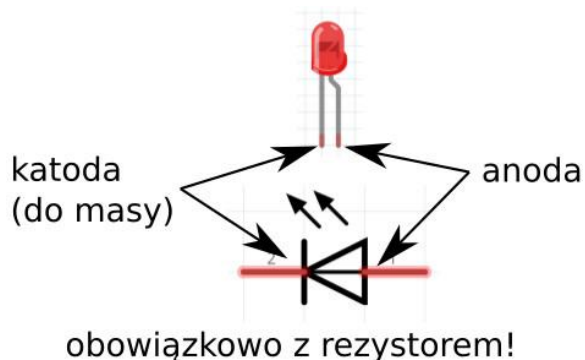
- jaki prąd popłynie, kiedy włączę zasilanie?
- jaki prąd popłynie, kiedy ustawię dane GPIO jako wyjście?
- jaki prąd popłynie, kiedy na wyjściu GPIO będzie stan wysoki / niski?

Nie używamy 5V, tylko 3V3.

Proszę używać przycisków i LED na adapterze do RPi.

Uwaga na przewody: mają prostokątne piny, które nie zawsze chcą się łatwo dopasować i łatwo się łamią.

Proszę nie wciskać na siłę, a w zamian spróbować połączyć pod innym kątem.



Uwaga: do podłączenia LED należy zastosować właściwy rezystor, który ograniczy prąd (należy obliczyć wartość rezystancji). Dotyczy to podłączania dodatkowych LEDów. Wszystkie LEDy na adapterze do RPi są wyposażone w rezystory.

Uwaga: do połączenia GPIO z liniami sygnałowymi akcesoriów można zastosować niewielki rezystor, np. 100 om, aby ograniczyć prąd, w razie popełnienia błędu (błędem jest np. ustawienie GPIO jako wyjście i podłączenie do źródła zasilania lub do masy). Takie rozwiązanie jest możliwe dla sygnałów cyfrowych o niskiej częstotliwości.

Obsługa GPIO przez sysfs

W systemie plików katalog /sys pozwala na dostęp do urządzeń za pomocą interfejsu sysfs.

W szczególności urządzenia GPIO są dostępne w katalogu:

/sys/class/gpio

Pliki w ww. katalogu to m.in.:

export
unexport

Wpisanie numeru przez export uruchomi obsługę danego pinu przez sysfs:

```
echo 27 > /sys/class/gpio/export
```

(powyższe polecenie oznacza: wpisz tekst "27" do pliku /sys/class/gpio/export, co spowoduje zarezerwowanie GPIO 27 do dalszych działań). Do GPIO jest podłączony LED 1 na adapterze do RPi.

Pojawi się katalog:

```
/sys/class/gpio/gpio27
```

i pliki w tym katalogu, m.in.: direction, edge i value

Kierunek (we / wy) można zmieniać wpisując do pliku:

```
/sys/class/gpio/gpio27/direction
```

jedną z wartości: in, out, low, high

gdzie:

- in - ustawia dane GPIO w tryb wejścia
- out - ustawia dane gpio w tryb wyjścia
- low - ustawia dane gpio w tryb wyjścia z początkową wartością niską
- high - ustawia dane gpio w tryb wyjścia z początkową wartością wysoką

Przykład:

```
echo out > /sys/class/gpio/gpio27/direction
```

Zgłaszanie przerwań przy zmianie stanu jest obsługiwane plikiem:

```
/sys/class/gpio/gpio27/edge
```

Odczyt bieżącej wartości dla wejścia i wyjścia:

```
cat /sys/class/gpio/gpio27/value
```

Ustawienie wartości dla wyjścia, np.:

```
echo 1 > /sys/class/gpio/gpio27/value
```

Wpisując na zmianę wartość 0 i 1 zaobserwujemy miganie LED 1 na adapterze do RPi.

Po zakończeniu pracy z danym GPIO przez sysfs należy zwolnić zasoby korzystając z pliku unexport, np.:

```
echo 27 > /sys/class/gpio/unexport
```

Obsługa GPIO w Pythonie

Na laboratorium 2 zalecamy pisanie programów w Pythonie.

W OpenWRT dostępny jest pakiet Pythona gpio4 (<https://pypi.org/project/gpio4>), który można zainstalować przez pip:

```
pip install gpio4
```

UWAGA: wcześniej należy zainstalować Python oraz pip za pomocą menedżera pakietów opkg.

UWAGA: W module gpio jedyną działającą poprawnie klasą jest SysfsGPIO. To z niej należy korzystać. Klasa ta zapewnia dostęp do GPIO za pośrednictwem sysfs.

Przykład użycia:

```
import gpio4
gpio27 = gpio4.SysfsGPIO(27)
gpio27.export = True          # use the pin
gpio27.direction = 'out'     # set direction to output
gpio27.value = 0             # set value to low
gpio27.value = 1             # set value to high
gpio27.export = False        # cleanup
```

Stanowisko pracy

Należy stosować się do instrukcji BHP. Praca w bałaganie może skutkować powstaniem zwarcia i awarią sprzętu!

Na pół godziny przed końcem zajęć przerywamy pracę i zaczynamy sprzątanie.

Zadania - obsługa akcesoriów

Do zadań 1, 2 i 3 nie trzeba robić schematu, gdyż można skorzystać z LED i przycisków na adapterze do RPi. Schemat należy przygotować dla zadań 4 i 5, w programie *fritzing*.

Zadanie 1: GPIO - wyjście dla LED

Napisz program `gpio_led_1.py`, który 10 krotnie włączy i wyłączy LED na adapterze do RPi

Zadanie 2: GPIO - wyjście dla LED z płynną zmianą jasności

Napisz program `gpio_led_2.py`, który przez 10 sekund będzie w płynny sposób zmieniał jasność LED na adapterze do RPi (np. funkcją `sin`, z wykorzystaniem sygnału modulowanego ze zmiennym wypełnieniem). Program ma generować sygnał PWM o stałej częstotliwości i zmiennym wypełnieniu w zakresie 0 - 100%.

Zadanie 3: GPIO - wejście

Napisz program `gpio_in.py`, który będzie reagował na wciśnięcie przycisku na adapterze do RPi, np. włączy LED na adapterze.

Zadanie 4: GPIO - wyjście PWM, buzzer pasywny

a) Przygotuj schemat podłączenia buzzera pasywnego do GPIO

b) Podłącz według schematu

c) Napisz program `gpio_snd.py`, który będzie generował kolejne dźwięki gamy C-dur w 2 oktawach.

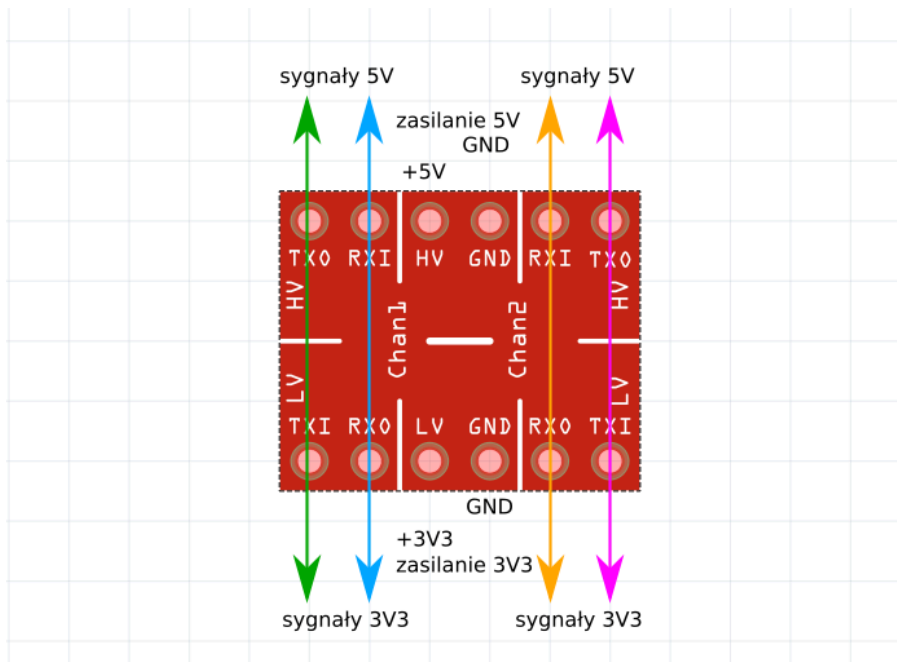
Program ma generować sygnał PWM o zmiennej częstotliwości i stałym wypełnieniu 50%.

UWAGA: Programowa implementacja PWM w Pythonie ma ograniczoną częstotliwość na stosunkowo niskim poziomie. Należy zmierzyć maksymalną częstotliwość PWM jaką można uzyskać. W tym celu należy zastosować w Pythonie funkcje do pomiaru czasu.

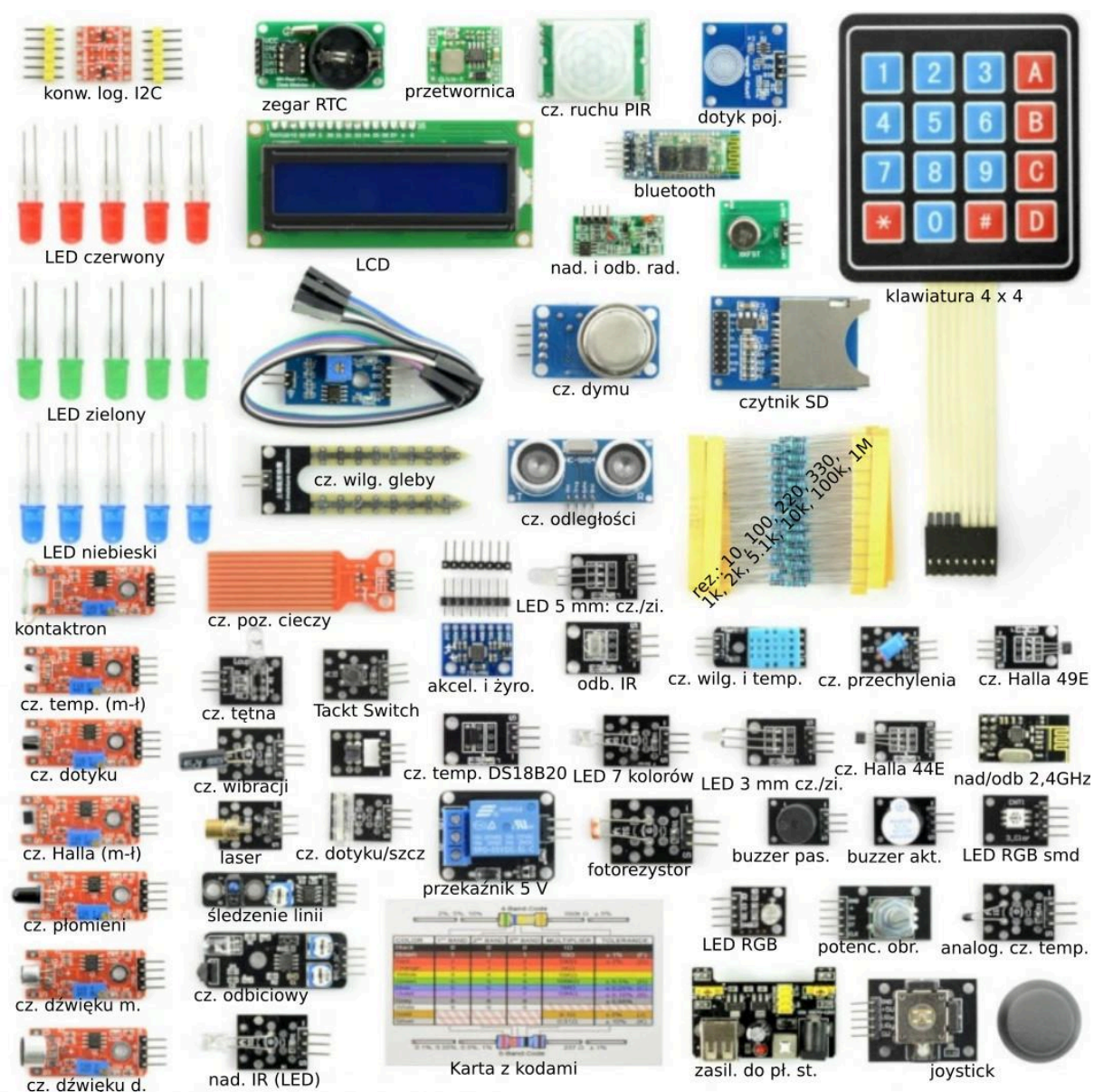
Zadanie 5: Akcesoria do wyboru

Proszę wybrać sobie czujnik lub efektor, który można obsłużyć przez GPIO (np. czujnik odległości, serwomechanizm) i napisać program, który go obsłuży.

UWAGA: w przypadku korzystania z urządzeń działających na 5V, należy skorzystać z konwertera poziomów logicznych, gdyż RPi działa na 3,3V. Układ ten nazywa się "logic level converter" w *fritzing* i należy go podłączyć w następujący sposób:



Akcesoria



Dodatkowo:

- przetwornik analogowo-cyfrowy 4-kanalowy - moduł DFR0316 (MCP3424)
- przetwornik cyfrowo-analogowy 4-kanalowy - moduł Adafruit P4470 (MCP4728)
- serwomechanizm analogowy 360 st. sterowany prędkościowo - PowerHD AR-3606HB

Dokumentacja do (większości) akcesoriów jest dostępna na Moodle.

Niektóre elementy nie mają zastosowania na laboratorium i nie są udostępnione, np. czujnik dymu, czujnik wilgotności gleby, czytnik kart SD, przełącznik 5V/250V, zasilacz do płytki stykowej.

Uwaga: część dokumentacji, którą otrzymaliśmy od dystrybutora jest bardzo niskiej jakości (słabe tłumaczenie z jeszcze słabszej chińskiej dokumentacji). Proszę podchodzić z pewnym dystansem i ograniczonym zaufaniem do tej do dokumentacji!

Uwaga: należy natychmiast poinformować prowadzącego o wszelkich usterkach sprzętu i błędach w dokumentacji i dostępnych materiałach.

Literatura

Dokumentacja do modułu Pythona gpio4:

<https://pypi.org/project/gpio4>

Uwaga: w przykładowych kodach są błędy - proszę wzorować się na kodach w niniejszej instrukcji.

Tutorial do serwomechanizmów modelarskich:

<https://learn.sparkfun.com/tutorials/hobby-servo-tutorial/all>

Dokumentacja do RPi:

<https://www.raspberrypi.com/documentation>

Podłączenie LED:

https://en.wikipedia.org/wiki/LED_circuit

Polecenie losetup:

<https://man7.org/linux/man-pages/man8/losetup.8.html>

Dokumentacja poleceń:

<https://man7.org/linux/man-pages>