

KAmoDRPi ADC DAC (PL)



Spis treści

Wymagania	1
Sposób podłączenia	2
Konfiguracja	3
Kod programu w Pythonie	4
Do pobrania	6
Linki zewnętrzne	7

Opis

[KAmoRRPi ADC_DAC](#) to moduł z 10-bitowym przetwornikiem analogowo-cyfrowym MCP3021 oraz z 10-bitowym przetwornikiem cyfrowo-analogowym MCP4716. Oba układy wykorzystują magistralę I2C. Zakres napięcia wejściowego przetwornika ADC ustalany może być za pomocą zworki; dostępne są trzy przedziały napięcia: 0 - 3,3 V ; 0 - 5 V; 0 - 10 V. Złącze przelotowe pozwala na dołączenie modułu do dowolnej wersji Raspberry Pi. Przykładowy program pokazuje, jak można obsłużyć moduł na komputerze Raspberry Pi, przy pomocy języka Python.

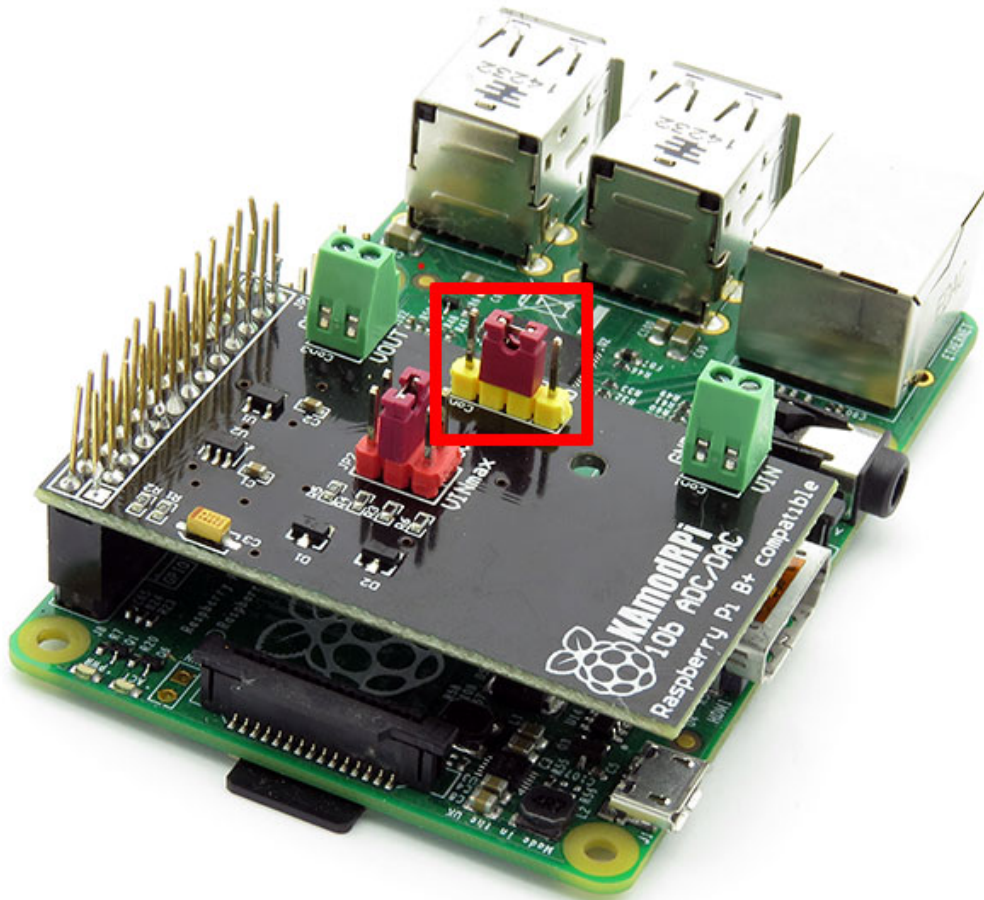
Wymagania

Do uruchomienia przykładowego kodu potrzebne będą:

- Raspberry Pi (dowolna wersja)
- Karta z systemem operacyjnym Raspbian
- Moduł KAmoRRPi ADC_DAC
- Zwórka lub przewód

Sposób podłączenia

- Nałożyć moduł KAmoDRPi ADC_DAC na Raspberry Pi
- Połączyć piny VIN oraz VOU zworką lub przewodem



Konfiguracja

Pierwszą czynnością będzie uruchomienie obsługi I2C w Raspbianie; w tym celu wykonujemy komendę:

```
sudo raspi-config
```

W oknie konfiguracji wybieramy kolejno `//Advanced Options//`, `//I2C//`, `//Yes//` (dwukrotnie), po czym uruchamiamy ponownie Raspberry Pi. Następnie, zainstalować musimy `i2c-tools` komendą:

```
sudo apt-get install i2c-tools
```

Sprawdzimy, czy Raspberry Pi może połączyć się z modułem poprzez I2C:

```
i2cdetect -y 1
```

Rezultat komendy powinien wyglądać następująco:

```

    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  4d  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60: 60  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --

```

Jak widać, komputer odnalazł dwa adresy urządzeń I2C - rozpoznajmy je:

- 0x4d - adres układu MCP3021
- 0x60 - adres układu MCP4716

Kod programu w Pythonie

Do obsługi magistrali I2C w Pythonie musimy zainstalować bibliotekę smbus: `sudo apt-get install python-smbus`. Mając zainstalowaną tą bibliotekę, umieścimy poniższy kod w pliku KAmoDRPiADCDAC.py:

```
import smbus

class KAmoDRPiADCDAC:

    VINmax = 10
    bus = smbus.SMBus(1)
    adAddress = 0x4d
    daAddress = 0x60

    def setVINmax(self, v):
        self.VINmax = v

    def readMCP3021(self):
        rd = self.bus.read_word_data(self.adAddress, 0)
        return ((rd & 0xFF) << 8) | ((rd & 0xFF00) >> 8)

    def readMCP3021Voltage(self):
        val = self.readMCP3021() >> 2
        return float(self.VINmax) * float(val) / float(0x3FF)

    def writeMCP4716(self, level):
        self.bus.write_i2c_block_data(self.daAddress, (level & 0x3C0) >> 6, [(level & 0x3F) << 2])
```

Możemy teraz utworzyć prosty program, który ustawi poziom wyjścia na wyjściu przetwornika DAC na wartość, którą zmieniać będziemy kursorami (góra/dół). Jednocześnie, program wyświetli wartość z wejścia przetwornika ADC:

```
import time
import sys
import curses
import KAmoDRPiADCDAC

screen = curses.initscr()
curses.noecho()
curses.cbreak()
screen.keypad(True)

try:
    KAmod = KAmoDRPiADCDAC.KAmoDRPiADCDAC()

    # Ustawienie VINmax na tą samą wartość, co zworka JP2 na płytce KAmod
    KAmod.setVINmax(5)

    daValue = 0
    while True:

        # Zapisz daValue do przetwornika DAC
        KAmod.writeMCP4716(daValue)

        # Odczytaj voltage z przetwornika ADC
```

```
voltage = KAmod.readMCP3021Voltage()  
screen.addstr('\rADC Voltage = %.2fV, DAC Value = %d      ' % (voltage,  
daValue))
```

Do pobrania

- [File:Kamodrpiadcdac.zip](#)

Linki zewnętrzne

- [Artykuł na portalu MIKROKONTROLER.PL: KAmoDRPI 10b ADC/DAC – ekspander z przetwornikiem ADC dla komputerów Raspberry Pi – Przykład w C++](#)



Zastrzegamy prawo do wprowadzania zmian bez uprzedzenia.

Oferowane przez nas płytki drukowane mogą się różnić od prezentowanej w dokumentacji, przy czym zmianom nie ulegają jej właściwości użytkowe.

BTC Korporacja gwarantuje zgodność produktu ze specyfikacją.

BTC Korporacja nie ponosi odpowiedzialności za jakiegokolwiek szkody powstałe bezpośrednio lub pośrednio w wyniku użycia lub nieprawidłowego działania produktu.

BTC Korporacja zastrzega sobie prawo do modyfikacji niniejszej dokumentacji bez uprzedzenia.