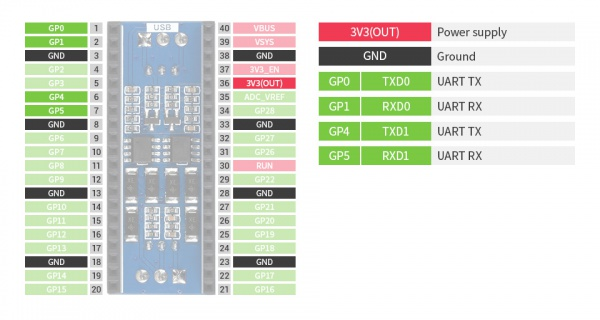
# RS485 - UART

## Waveshare UART-to-RS485 2ch HAT for Pico

Nietypowy, ale wygodny konwerter, z wbudowanym automatycznym kierunkiem transmisji. Nie potrzeba jawnie sterować sygnałem RE/DE.



from machine import Pin, UART

import utime

uart = UART(0, baudrate=9600, tx=Pin(0), rx=Pin(1))

#uart = UART(1, baudrate=9600, tx=Pin(4), rx=Pin(5))

while True:

uart.write('Hello!\n')

print('\*', end='')

if uart.any():

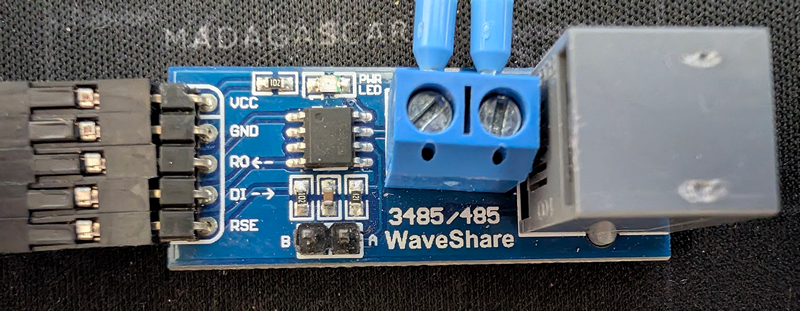
print('Rx:', uart.read())

utime.sleep(1)

## 

## Tani konwerter UART-RS485 z sygnałami kierunku

Większość dostępnych tanich konwerterów, np.:



from machine import Pin, UART

import utime

#uart = UART(1, baudrate=9600, tx=Pin(4), rx=Pin(5))

uart = UART(0, baudrate=9600, tx=Pin(0), rx=Pin(1))

rse = Pin(2, Pin.OUT)

while True:

rse.value(1)

uart.write('Hello!\n')

print('\*', end='')

utime.sleep(0.1)

rse.value(0)

utime.sleep(0.1)

if uart.any():

print('Rx:', uart.read())

utime.sleep(1)

## Konwerter RS485-USB (PC, Raspberry Pi)

Może być dowolny, dobrze aby miał sygnalizację aktywności TXD i RXD, np.:



Ten ze zdjęcia jest na bazie FT232, co oznacza niezłą jakość sterowników i stabilne działanie.

Uwaga! Jest też tańsza wersja B, która też działa, ale nie jest to FT232.

## Konwerter RS485 wbudowany na płycie BeamKit

* DI - Data input
* RO - Receiver Output
* DE/nRE - direction
* D42 - Rx
* D43 - Tx
* TERM - enable termination resistor 120 Ohm
* A=A, B=B (same signals)

