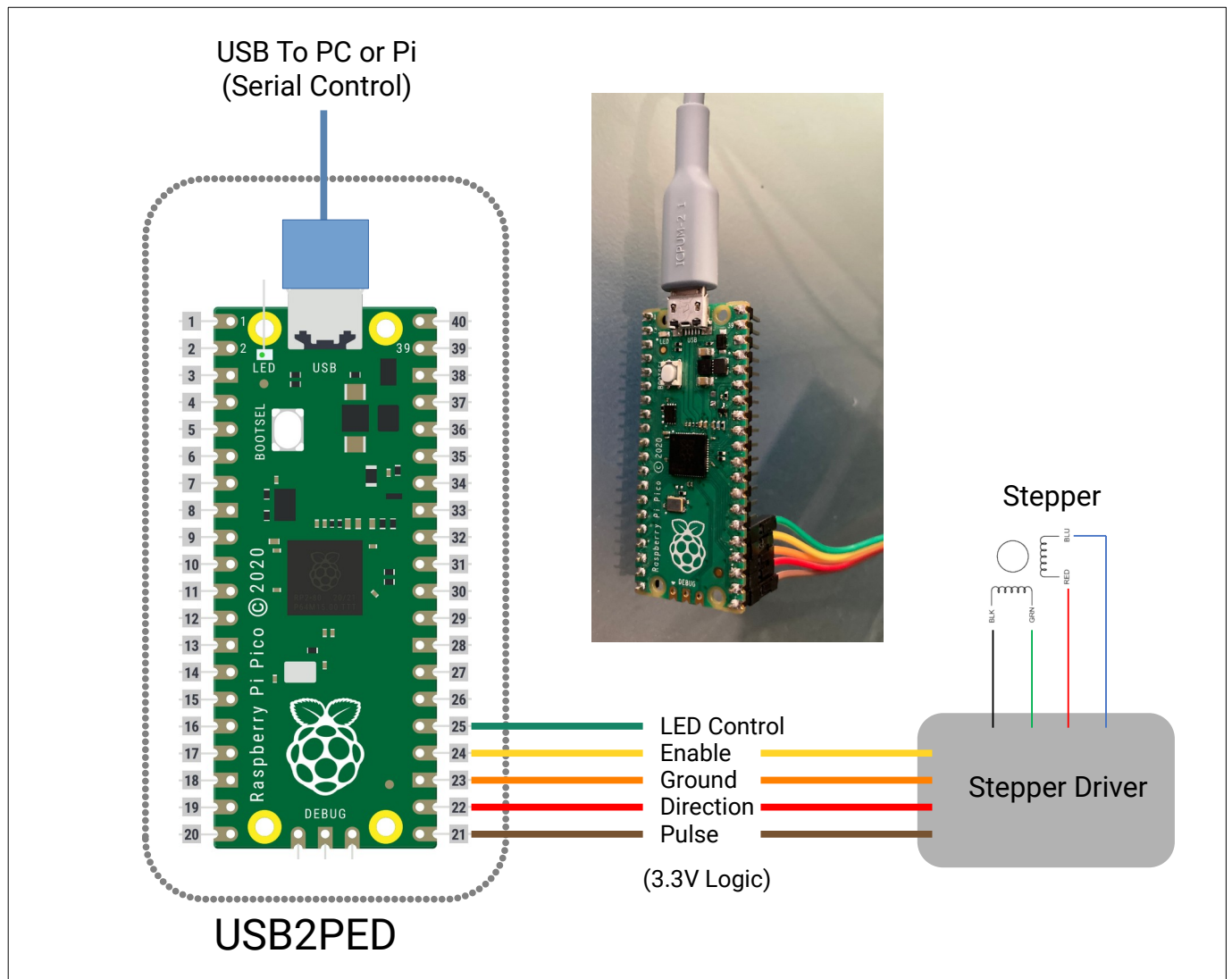


PICO_USB2PED

GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007
Version 1.0 by PM490

Code and information provided as-is, no warranties whatsoever.

Code to use Raspberry PICO as interface converter from USB-Serial to Pulse-Enable-Direction Stepper Driver and LED Control. This is useful in DIY telecine/film scanners controlling the frame movement with an stepper motor.
Repository includes the file PICO_USB2PED.uf2 to flash/upload to PICO.



Tutorial

Follow these steps to flash the PICO:

- Download the file PICO_USB2PED.uf2 at the flash folder in the repository.
- Press and hold the Bootsel button (button next to the usb connector) then connect the pico to the PC (while holding it pressed). After is connected, let go of the bootsel button.
- The pico will show in your PC or RPi as a storage device (similar to a usb-drive).
- Copy the file PICO_USB2PED.uf2 you downloaded to the storage device. That will flash the PICO. The storage device will then disappear, and the PICO is now programmed.
- To test manually, any serial terminal can communicate with the added Serial port (PICO connected to a USB port). Check what serial port was assigned by the PC or RPi. In the RPi it typically is assigned /dev/ttyACM0.

For the Terminal program (Putty Serial for example) use the following settings to connect:

Port = /dev/ttyACM0 for Rapsberry or COM port for other OS.
baudrate = 115200
bytesize = 8
stopbits = 1
parity = N

Once you are able to connect via the terminal program, typing the letter-command will allow you to see/test the results on the corresponding GPIO pins.

L - will make GP19 (Led) High.

I - will make GP19 (Led) Low.

E - will make GP18 (Enable) High.

e - will make GP18 (Enable) Low.

D - will make GP17 (Direction) High.

d - will make GP17 (Direction) Low.

T - will send the one-turn number of pulses out of GP16 (Pulse). See the PICO_USBPED

See additional commands to set/change number of pulses per turn and frequency of the pulses.

Control Characters

	Char	Hex
Pulse Output	P	50
Pulse Off	p	70
Enable High	E	45
Enable Low	e	65
Direction High	D	44
Direction Low	d	64
Turn	T	54
LED High	L	4C
LED Low	I	6C

Set Pulses per Turn

Pulses/Turn	Char	Hex
200	0	0x30
400	1	0x31
800	2	0x32
1,600	3	0x33
3,200	4	0x34
6,400	5	0x35
12,800	6	0x36
25,600	7	0x37
51,200	8	0x38

Frequency Selection

Frequency (Khz)	Period (uSec)	Char	Hex
7.94	126	*	2A
7.04	142	&	26
6.02	166	^	5E
5.00	200	%	25
4.00	250	\$	24
2.99	334	#	23
2.00	500	@	40

Controlling with Raspberry Pi or PC

Python Code Example

```
# Library
import serial
import time

# Set serial for pico-transport
ser = serial.Serial()
ser.port = '/dev/ttyACM0'
ser.baudrate = 115200
ser.bytesize = 8
ser.stopbits = 1
ser.parity = 'N'
ser.timeout = 2
ser.rtscts = 0
ser.open()

# Read PICO Startup Line
received_startup = ser.readlines()

# Command Format – Example: LED ON for 2 seconds
ser.write(b'L') #Set LED pin to High
time.sleep(2)
ser.write(b'l') #Set LED pin to Low

# Command Format – Example: Set Direction, and make one turn
ser.write(b'D') #Set Direction pin to High
ser.write(b'E') #Set Enable pin to High
ser.write(b'T') #Send Pulses for One Turn.
time.sleep(1)   #Wait for Turn to complete
                #The turn may take longer,
                #set delay based on quantity of pulses and frequency.
ser.write(b'e') #Set Enable pin to High
```

PINOUT

Wiring PICO to Stepper Driver & LED

PICO		PICO			MODE	Function
GP0	1	40	VBUS			
GP1	2	39	VSYS			
GND	3	38	GND			GND
GP2	4	37	3V3_EN			
GP3	5	36	3V3 (OUT)			
GP4	6	35	ADC_VREF			
GP5	7	34	GP28			
GND	8	33	AGND			
GP6	9	32	GP27			
GP7	10	31	GP26			
GP8	11	30	RUN			
GP9	12	29	GP22			
GND	13	28	GND			GND
GP10	14	27	GP21			
GP11	15	26	GP20			
GP12	16	25	GP19	OUT		LED Control
GP13	17	24	GP18	OUT		ENABLE
GND	18	23	GND			GND
GP14	19	22	GP17	OUT		DIRECTION
GP15	20	21	GP16	OUT		PULSE