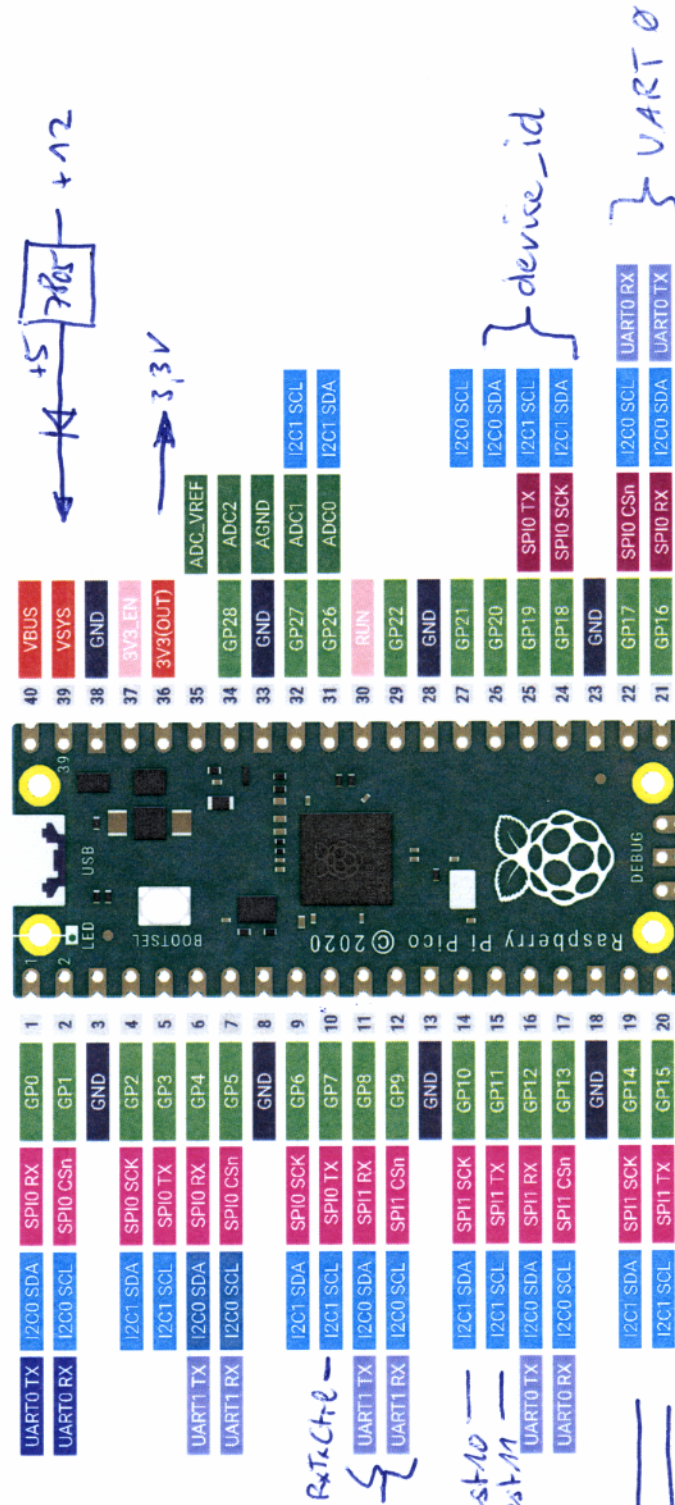


# Raspberry Pi Pico Pinout

UART 0 → WR<sub>intcon</sub>  
 UART 1 → WR ↔ CTRL

Power
Ground
UART / UART (default)
GPIO, PIO, and PWM
ADC
SPI / SPI (default)
I2C / I2C (default)
Reserved
Debugging

LED (GP25)



SWDIO  
 GND  
 SWCLK

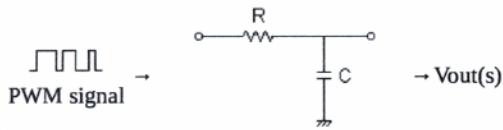


Raspberry Pi

## RC Low-pass Filter Design for PWM - Result -

Calculated peak-to-peak ripple voltage and settling time at a given PWM frequency and cut-off frequency or values of R and C.

### RC Filter



Transfer Function:

$$G(s) = \frac{1}{s+1}$$

$f_{\text{PWM}} = 120$  Hz

Duty Step 0% → 90 [%]

PWM signal voltage:

$V_L = 0$  [V]  $V_H = 3$  [V]

R and C values of filter | Cut-off frequency

☐ Cut-off frequency  $f_c = 1k$  [Hz]

☒ R and C values

$R = 10k$   $\Omega$   $C = 100u$  F

### Cut-off frequency

$f_c = 0.1591549430919$  [Hz]

p:pico, n:nano, u:micro, k:kilo, M:mega

### Final Vout value of the step response (without a ripple)

$g(\infty) = 2.7$  [V]

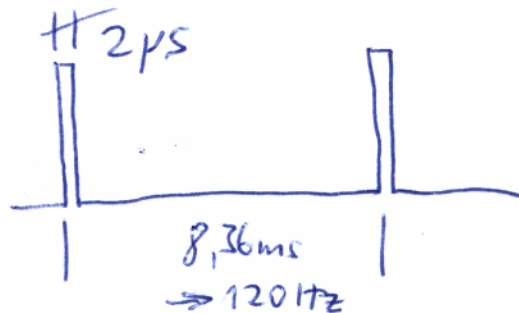
### Peak-to-peak ripple voltage

$\Delta V_{\text{pk-pk}} = 0.0062499909577703$  [V] (Duty=50%)

### Settling time 0% → 90% (0V → 2.43V) (without a ripple)

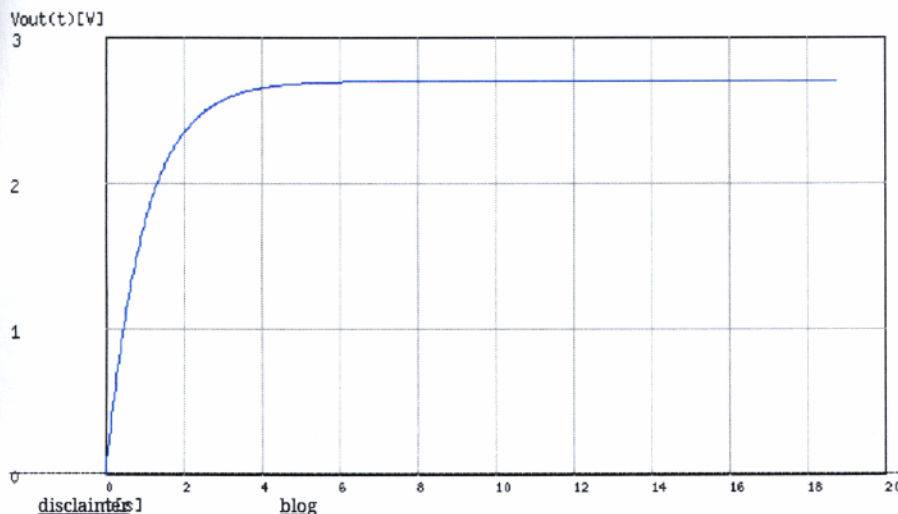
$= 2.302585092994$  [sec]

Calculate



## Transient analysis

StepResponse



[disclaimer](#)

[blog](#)

# RS485 Pin-Belegung

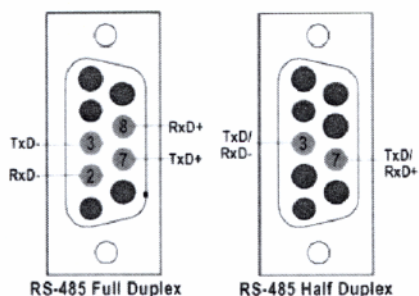
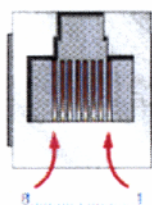


Table A-10 Pin Assignments for the S7-200 Communications Port

Connector	Pin Number	PROFIBUS Signal
	1	Shield
	2	24 V Return
	3	RS-485 Signal B
	4	Request-to-Send
	5	5 V Return
	6	+5 V
	7	+24 V
	8	RS-485 Signal A
	9	Not applicable
Connector shell		Shield

404

View from underneath



Pin	Signal
1	CANopen signal reserved
2	CANopen signal reserved
3	CANopen signal reserved
4	D1 Modbus signal
5	D0 Modbus signal
6	Not connected
7	VP(+)
8	Common Modbus signal



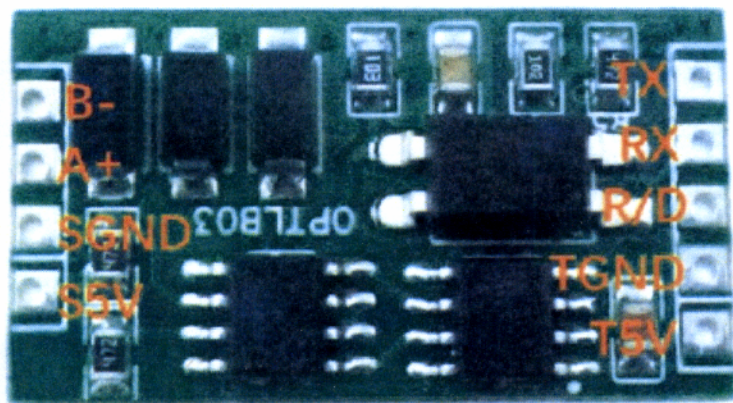
Table A-10 Pin Assignments for the S7-200 Communications Port

Connector	Pin Number	PROFIBUS Signal
	1	Shield
	2	24 V Return
	3	RS-485 Signal B
	4	Request-to-Send
	5	5 V Return
	6	+5 V
	7	+24 V
	8	RS-485 Signal A
	9	Not applicable
Connector shell		Shield

404

(1) Supply for RS232 / RS485 converter or a remote terminal





## Pin description

**S5V** : RS485 Power+

**SGND**: RS485 Power-

**A+** : RS485 A+

**B-** : RS485 B-

**T5V** : TTL Power+

**TGND**: TTL Power-

**R/D** : Data flow control signal

(Usually set to low level, MCU send data is set to high level)

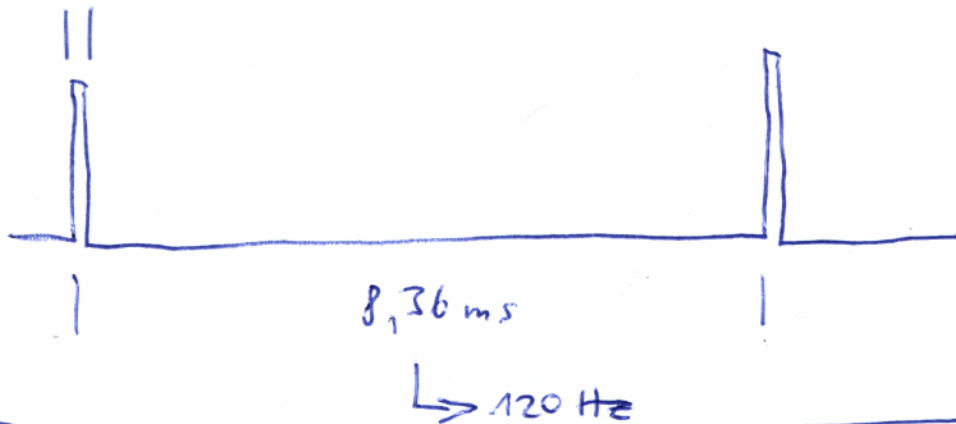
**RX** : TTL RXD(Connect MCU RXD pin)

**TX** : TTL TXD(Connect MCU TXD pin)

0x95			
[278]	1	09	} 237.4 V $V_{U_L}$
279	2	46	
280	3	0	
281	4	0	} 0 P <sub>out</sub>
282	5	0	
283	6	23	} 4.525 V $U_{B_{th}}$
284	7	2	
285	8	5	} Temperature
286	9	1	
287	10	2	
288	11	0	
289	12	4	
290	13	3	
291	14	EC	

2µs

PWM on Raspberry Pico



120 Hz

response Tx1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	
2B	00	B1	0E	09	06	00	00	00	22	01	91	00	00	00
Start byte		Voltage 231.0		Power 0		Battery 1.4V		Temperature		Edge		Error 6702		
Device ID		Response		Length		CRC								

request Tx1

2B	00	B0	02	02	02	E1
----	----	----	----	----	----	----

Req 01 03 00 3C 00 01 44 06 | 0D 04

Resp 01 03 02 00 00 B8 44 00

2 byte data 4 byte data CRC

+40ms

Req 01 03 00 3F 00 02 F4 07 | 0D 04

Resp. 01 03 04 1A 2E 00 00 9D 22 00

low High data CRC

+40ms

4 byte data