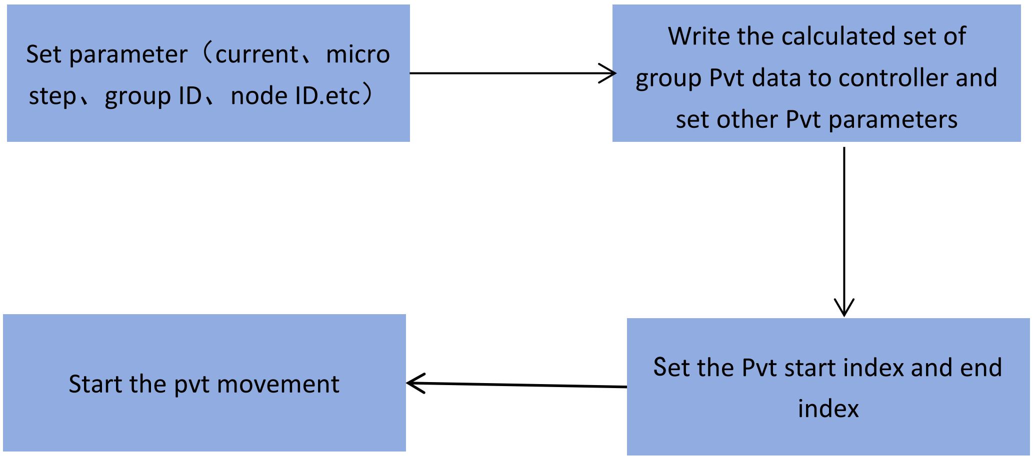


The process of PVT mode

Note: When using Pvt mode 3, there is no need to set the index.

Parameters to be set

General parameters

Node ID: The station number of the slave, which should not be duplicated

Object name	Node ID
SDO ID	0x2002
Object type	U8, rw
Range	1-127
Storage type	ROM
Default value	5

Micro step

Object name	Micro step
SDO ID	0x600A
Object type	U16, rw
Range	0,2,4,8,16,32,64,128,256
Storage type	ROM
Default value	0

Current

Object name	Maximum phase current
SDO ID	0x600B
Object type	U16, rw

Range	0-6000
Storage type	ROM
Default value	0

Group ID

Object name	Group ID
SDO ID	0x2006
Object type	U8, rw
Range	1-127
Storage type	ROM
Default value	0

Working mode: Switch to 2, Pvt mode

Object name	Working mode
SDO ID	0x6005
Object type	U8, rw
Range	0,1,2
Storage type	RAM
Default value	0

PVT Parameters

Pvt working mode[6010][02]

Subindex 0x02: PVT working mode

Object type	U8, rw
Range	0-2
Default value	0

0: PVT mode 1
1: PVT mode 2
2: PVT mode 3

Pvt points[6010][03]

Subindex 0x03: maximum PVT points

Object type	U16, rw
Range	0-1000
Default value	0

Pvt position[6010][11]

Subindex 0x11: PVT position

Object type	S32, rw
Range	-2147483648-2147483647
Default value	0

The absolute position to which the current PVT point is expected to move

Pvt speed[6010][12]

Subindex 0x12: PVT speed

Object type	S32, rw
Range	-2147483648-2147483647
Default value	0

The speed to which the current PVT point is expected to reach, in pps.

Pvt time[6010][13]

Subindex 0x13: PVT time

Object type	S32, rw
Range	0-2147483647
Default value	0

The time from the last PVT point to the current PVT point, in ms.

Pvt control[6010][01]

Subindex 0x01: PVT control

Object type	U8, rw
Range	0-3
Default value	0

0: stop PVT movement

1: start PVT movement

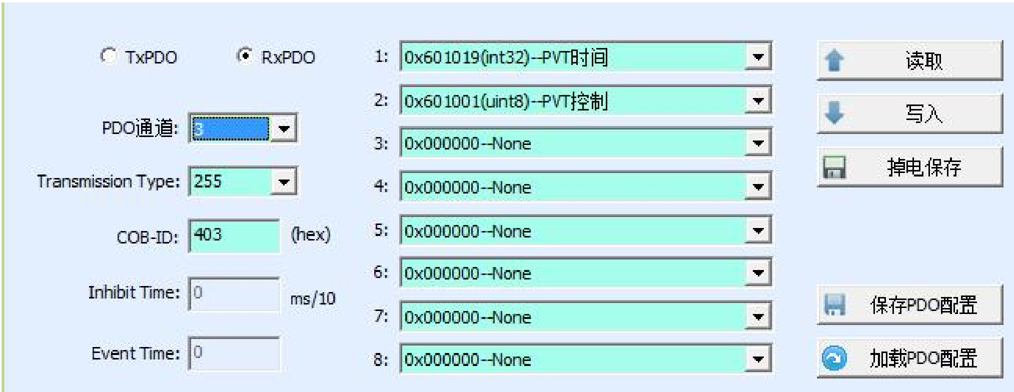
2: PVT position, velocity, and time object data are written to the queue

3: Clear all PVT data in the queue

Note: After the PVT parameters are written, it is also necessary to write 2 through [6010] [01] to write each set of parameters into the PVT queue, otherwise the PVT motion motor will not move when it is started.

Example

Taking Pvt mode 1 as an example, the trapezoidal acceleration motion is completed. Since this is a PDO communication method, the RPDO mapping is set up first. PDO mapping can be easily configured using our Pusican debugging software (when configuring multiple configurations, you can use the functions of saving PDO configuration and loading PDO configuration).



The following is a sample script:

Firstly, the algorithm of parameters in Pvt motion is designed, and the main variables are:

```
--**************************主程序******************************
--要求RPDO通道2设置位置速度映射, 通道3设置时间控制映射
local groupid = 55 --组id
local accel = 16000 --加速度
local decel = 16000 --减速度
local start_v = 600 --启动速度
local end_v = 600 --停止速度
local max_v = 32000 --最大速度
local time_interval = 100 --每一次运行时间

local curr_position --目标位置
local curr_velocity --目标速度
local curr_time --当前时间
local last_position --上一阶段的最终位置
local last_velocity  --上一阶段的最终速度
```

Pvt parameters are calculated and written in the uniform acceleration phase

```
--*******************first accel*******************
--[[
    s = v0*t + a*t*t/2
    v = v0 + a*t
--]]

for i=1,100
do
    curr_time = (i - 1)*time_interval/1000
    curr_position = start_v*curr_time + accel*curr_time*curr_time/2
    curr_velocity = start_v + accel*curr_time
    position_wr = math.floor(curr_position+0.5) --取整四舍五入
    velocity_wr = math.floor(curr_velocity+0.5)
    print("PVT"..i.."("..position_wr..","..velocity_wr..","..time_interval..")")
    f:write(position_wr..","..velocity_wr..","..time_interval.."\n")
    write_pvt(5,position_wr,velocity_wr,time_interval)  --写入本组pvt数据
    pusi.sleep(5)
    pt_idx = pt_idx + 1
    if curr_velocity >= max_v then
        break
    end
end
```

Constant velocity phase

```
last_position = curr_position
for i=1,30
do
    curr_time = i*time_interval/1000
    curr_position = last_position + curr_velocity*curr_time
    position_wr = math.floor(curr_position+0.5)
    velocity_wr = math.floor(curr_velocity+0.5)
    print("PVT"..i.."("..position_wr..","..velocity_wr..","..time_interval..")")
    f:write(position_wr..","..velocity_wr..","..time_interval.."\n")
    write_pvt(5,position_wr,velocity_wr,time_interval) --写入本组pvt数据
    pusi.sleep(5)
    pt_idx = pt_idx + 1
end
```

Uniform deceleration phase

```
--*******************third decel*******************
last_position = curr_position
last_velocity = curr_velocity
for i=1,100
do
    curr_time = i*time_interval/1000
    curr_velocity = last_velocity - decel*curr_time
    curr_position = last_position + last_velocity*curr_time - decel*curr_time*curr_
    position_wr = math.floor(curr_position+0.5)
    velocity_wr = math.floor(curr_velocity+0.5)
    print("PVT"..i.."("..position_wr..","..velocity_wr..","..time_interval..")")
    f:write(position_wr..","..velocity_wr..","..time_interval.."\n")
    write_pvt(5,position_wr,velocity_wr,time_interval) --写入本组pvt数据
    pusi.sleep(5)
    pt_idx = pt_idx + 1
    if curr_velocity <= end_v then
        break
    end
end
```

Next, set the start and end indexes

```
pvt1_start_idx = 0
pvt1_end_idx = pt_idx - 2
--PVT1起始索引
set_pvt1start(5,pvt1_start_idx)
--PVT1结束索引
set_pvt1end(5,pvt1_end_idx)
```

Note: pt_idx is a count variable, the initial value is 1, and the index starts from 0, so the

termination index is pt_idx the last value minus 2.
Finally, start the Pvt campaign.

```
--启动步进
start_pvt_step(groupid)
```

The start step here is in the form of NMT broadcasting, which will send commands to the slaves under the same set of IDs.

5.12 PV/PVT Synchronous start and stop

PMC007 can achieve two or more nodes PVT synchronous start/stop in a network by NMT extended instructions of standard CANopen.

Standard NMT format

COB-ID	Byte0	Byte1
0x000	CS	Node-ID

The extended NMT instructions add new definitions to Byte0 and Byte1 without affecting the standard protocol.

Byte0 is defined as follows:

Command	Function
10	Start PV motion
11	Start PVT motion
12	Stop PVT motion

Byte1 is group ID. The corresponding command operation is performed only when the group ID received by the controller matches its own group ID.

For example, when the group ID is set to 1.
To start a Pvt campaign should be sent: 000 0B 01
To stop the Pvt movement should be sent: 000 0C 01
For a single controller, writing 1 and 0 to the [6001][01] register can also be used to start and stop PVT motion.

FAQ about the Pvt model

1. Q: Does Pvt mode support variable acceleration movement?
A: Yes. Pusican comes with a sample script 3 that is a variable acceleration motion.

2. Q: Why won't the motor rotate after I set the position, speed, and time of the Pvt?
Answer: If you set the position, speed, and time of the Pvt, you need to write 2 to [6001][01] to write the parameters to the Pvt queue.

3. Q: Why is my position different from the target position after the exercise?
Answer: In PVT mode, the position is calculated by the number of subdivided pulses, and the position of the closed loop is calculated by the encoder scale, although the two are different, but the two have a fixed ratio that can be converted. For example, under 16 subdivisions, 10000 positions = 12 absolute encoder positions with 12 resolutions (10000/16/200*4096=12800).