7 .program ws2812 # defining a program named ws2812， all the assembled programs appear in the output file

8 .side\_set 1 # using 1 delay bit to “side-set”

9

10 .define public T1 2 # <=> #**define** T1 2 can be used for other software( with **public** keyword)

11 .define public T2 5 #

12 .define public T3 3 #

13

14 .lang\_opt python sideset\_init = pico.PIO.OUT\_HIGH # MicroPython PIO related, no need to worry about them

15 .lang\_opt python out\_init = pico.PIO.OUT\_HIGH #

16 .lang\_opt python out\_shiftdir = 1 #

17

18 .wrap\_target #

19 bitloop: # a **label** used with **jmp** instructions

20 out x, 1 side 0 [T3 - 1] ; Side-set still takes place when instruction stalls

# set 0 on the side-set pin. Shift 1 bit out of the OSR into x. wait T3-1 cycles then.

# x: scratch register(SM used to hold and compare temporary data), here is the destination of the write data

# [T3-1] number of delay cycles =T3-1

# side 0: drive low (0) the pin configured for side-set. ? set the configured pin to LOW

21 jmp !x do\_zero side 1 [T1 - 1] ; Branch on the bit we shifted out. Positive pulse

# side 1 on the side-set pin. if(x==0) goto do\_zero. Regardless of the condition, delay [T1-1] cycles

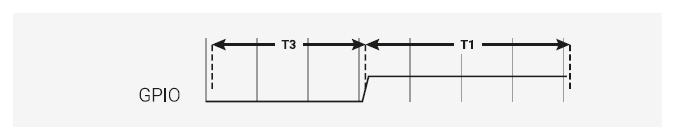


Figure : From the start of line 20 to then end of line 21

22 do\_one: # label used for jmp

23 jmp bitloop side 1 [T2 - 1] ; Continue driving high, for a long pulse # continue PULSE, jmp to bitloop

24 do\_zero: #

25 nop side 0 [T2 - 1] ; Or drive low, for a short pulse # side 0, no other loop, total execute T2 cycles

# nop: no operation. Defined as mov y,y in PIO assembly

26 .wrap # goto .wrap\_target immediately (0 cycle waste)

* Side-set: allows state machines to change the level or direction of up to 5 pins, concurrently with the

main execution of the instruction.

* OSR: staging area for data (TX FIFO -> SM-> OSR)