

Note: If a sequence is in use in a simple or complex sequence, it must have a length of `SEQ[n].MAXCNT > 0`.

```
NRF_PWM0->PSEL.OUT[0] = (first_port << PWM_PSEL_OUT_PORT_Pos) |
    (first_pin << PWM_PSEL_OUT_PIN_Pos) |
    (PWM_PSEL_OUT_CONNECT_Connected <<
        PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE = (PWM_MODE_UPDOWN_Up << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER = (PWM_PRESCALER_PRESCALER_DIV_1 <<
    PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
// Enable the shortcut from LOOPSDONE event to DMA.SEQ1.START task for infinite loop
NRF_PWM0->SHORTS = (PWM_SHORTS_LOOPSDONE_DMA_SEQ1_START_Enabled <<
    PWM_SHORTS_LOOPSDONE_DMA_SEQ1_START_Pos);
// LOOP_CNT must be greater than 0 for the LOOPSDONE event to trigger and enable looping
NRF_PWM0->LOOP = (1 << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER = (PWM_DECODER_LOAD_Common << PWM_DECODER_LOAD_Pos) |
    (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
// To repeat a single sequence until stopped, it must be configured in SEQ[1]
NRF_PWM0->DMA.SEQ[1].PTR = ((uint32_t)(seq0_ram)) << PWM_DMA_SEQ_PTR_PTR_Pos;
NRF_PWM0->DMA.SEQ[1].MAXCNT = (sizeof(seq0_ram) << PWM_DMA_SEQ_MAXCNT_MAXCNT_Pos);
NRF_PWM0->SEQ[1].REFRESH = 0;
NRF_PWM0->SEQ[1].ENDDELAY = 0;
NRF_PWM0->TASKS_DMA_SEQ[1].START = 1;
```