perplexity

Delay Calculation Basics in AVR Assembly

Let's break down how delay loops work in AVR microcontrollers—step by step—starting from the fundamentals.

1. What Is a Delay Loop?

- **Purpose:** It makes the microcontroller "wait" for a certain amount of time before proceeding. This is used to create things like blinking LEDs.
- **How:** The CPU repeats a set of instructions many times—a loop. Each time through, it adds a known, small time delay.

2. What You Need to Know to Calculate Delay

- **Given to you:**
  - Code of the delay loop (how many loops, what instructions)
  - **Crystal frequency** (how fast your chip runs, e.g., 16 MHz)
- **You must compute:**
  - How long each instruction takes.
  - How many times total the instructions are run in all loops.
  - Multiply total instruction count by the instruction cycle time to get REAL TIME delay.

3. How Do You Find the Delay from Code?

a) Step 1: Find the Clock Frequency

- ATmega chips typically use 8, 16, or 20 MHz crystals.
- The **cycle time** is

$$\text{Instruction cycle time} = \frac{1}{\text{Crystal Frequency}} \text{ seconds}$$

  - Example: For 16 MHz, one cycle is $ \frac{1}{16,000,000} = 0.0625 \mu s $[1]

b) Step 2: Count Instruction Cycles

- Each AVR instruction takes a set number of cycles (usually 1 cycle for simple instructions, 2 cycles for a conditional branch, 4 for RET).
- For each loop, count how many times each instruction runs.
- For a nested loop (loop inside a loop), multiply the passes:

  - Total times run = Outer × Middle × Inner

c) Step 3: Multiply Instructions × Cycles × Passes

- Add up cycles for each instruction per loop, multiply by loop counts:
  Example:

  ```
  MOV R18, 255    ; 1 cycle, runs once
  LOOP1:
    MOV R17, 255  ; 1 cycle, runs 255×
  LOOP2:
    DEC R17       ; 1 cycle
    BRNE LOOP2    ; 2 cycles if not zero, else 1 cycle
    DEC R18       ; 1 cycle
    BRNE LOOP1    ; 2 cycles if not zero, else 1 cycle
    RET           ; 4 cycles
  ```

- Calculate yourself:
  - Inner loop: (number of steps per run) × (times run)
  - Add startup and ending instructions.
- Finally: **Total cycles × Cycle time = Total delay in seconds**

4. Why Three Loops?

- AVR chips run FAST. One loop can only make a short delay (a few microseconds).
- Nesting loops multiplies the total delay:
  $$\text{Total Loops} = L_1 \times L_2 \times L_3$$
- More loops ⟶ Longer delay (can reach milliseconds or even seconds).
  - Example: For 1 second delay at 1 MHz, need about 1,000,000 cycles total.[2]

5. How Many Loops Do You Need?

- If the required number of cycles is low, one loop is enough.
- For longer delays, you need two or three nested loops so you can use small 8-bit registers and still get big counts.
- There’s no strict rule—the goal is to avoid register overflow and keep code simple.[3]

6. If Given a Delay Value, What Do You Compute?

Suppose you're told: "Create a delay of 0.5 seconds."

Steps:

1. **Calculate total cycles needed:**
   $$\text{Total cycles required} = \text{Desired time} \times \text{Crystal frequency}$$
   Example (0.5 s, 16 MHz): $0.5 \times 16,000,000 = 8,000,000$ cycles
2. **Design your nested loops to total about 8,000,000 cycles**

    - E.g., 250 × 250 × 128 = 8,000,000
    - Calculate cycles per loop and tweak loop counts. See examples above and.[4]
3. **Implement using LDI, DEC, BRNE instructions.**

7. Summary Table

Parameter	What it Means
Crystal frequency	Chip speed (Hz)
Cycle time	Time per instruction
Loop pass count	How many times each loop repeats
Instruction cycles	How many cycles each instruction uses
Total cycles	Product of all the above
Total delay	Total cycles × cycle time

8. Why Not Use Timers?

You want simple delays, so we use code loops. For precise or background delays, AVR timers and interrupts can do better (because loops "lock" the CPU).

Let me know if you'd like a worked example with your code or a delay loop of your choice! You can share any value and I can walk you through the calculation stepwise.

⁂

1. https://www.tutorialspoint.com/microprocessor/time_delay_in_avr_microcontroller.htm
2. https://stackoverflow.com/questions/24097526/how-can-i-make-a-delay-in-assembly-for-avr-microcontrollers
3. https://adc.anteh.ru
4. http://www.rjhcoding.com/avr-asm-delay-subroutine.php
5. https://www.geeksforgeeks.org/electronics-engineering/time-delay-in-avr-microcontroller/
6. https://www.youtube.com/watch?v=Zw5hf8nibAY
7. https://www.avrfreaks.net/s/topic/a5C3I000000U3bLEAS/t027643
8. http://darcy.rsgc.on.ca/ACES/TEI4M/AVRdelay.html
9. https://www.youtube.com/watch?v=L5JtVa853Z4