

13 April 2021

Code:

Module/Function	Calls	Time(Sec)	Time(%)
Inline		4.988 s	100%
Serial.c		4.427 s	89%
SER_Init	2	456.833 us	0%
\$Sub\$\$rt_entry	1	0.083 us	0%
SER_PutChar	76438	4.426 s	89%
SER_waitLess	0	0us	0%
SER_GetChar	0	0us	0%
generic_algorithm.c		553.580 ms	11%
print_to_uart.c		7.795 ms	0%
startup_stm32f10x_md.s		10.907 us	0%
Inline.c		4.769 us	0%
system_stm32f10x.c		2.019 us	0%
Retarget.c		0us	0%

Performance Analyzer			
<input type="button" value="Reset"/> Show: <input type="text" value="Modules"/>			
Module/Function	Calls	Time(S...)	Time(%)
[-] Inline		3.417 s	100%
[-] generic_algorithm.c		3.416 s	100%
[-] gen_algo	309	3.416 s	100%
[-] Serial.c		496.972 us	0%
[-] print_to_uart.c		100.139 us	0%
[-] Inline.c		28.685 us	0%
[-] startup_stm32f10x_md.s		10.907 us	0%
[-] system_stm32f10x.c		2.019 us	0%
[-] Retarget.c		0us	0%

Flag control

[illegible]

```
USER#  
What the hell did you just say about me, you little nose-swigger? I'll have you know I graduated top of my class in the Navy Service Academy.
```

The ***Flag control*** method sends ***one character*** at a time (keeping track of index via static int) from the message then executes the generic algorithm method that simulates the CPU tracking of the IP address. This method uses basically 0% of the execution time to send the message which leaves approximately 100% of the execution performance to the CPU. Making it a more efficient method to track the IP address but takes longer to send the message to the person on the other end of the connection.

A better method would be to run both the generic algorithm and sending message function in parallel. So that neither function must wait for the other to finish.