Calculation speed of an ATmega328-based Arduino running at 16 MHz

(including the time required to read and write to RAM - see note 1)

Operations	Byte (1 byte)		Int (2 bytes)		Long (4 bytes)		Float (4 bytes)	
	µs	CPU cycles	µs	CPU cycles	µs	CPU cycles	µs	CPU cycles
Read or write to RAM	**0.13**	**2**	**0.25**	**4**	**0.50**	**8**	**0.50**	**8**
Addition, z = x + y	0.44	7	0.88	14	1.76	28	*8.11*	*130*
Subtraction, z = x - y	0.44	7	0.88	14	1.76	28	*8.15*	*130*
Multiplication, z = x * y	0.63	10	1.50	24	4.53	72	*9.67*	*155*
Division, z = x/y	*12.9*	*207*	*15.7*	*251*	*40.9*	654	*31.1*	*497*
Modulo, z = x % y	*12.9*	*207*	*15.7*	*251*	*40.9*	654	*31.1*	*497*
Min and Max, z = min(x,y)	0.73	12	1.29	21	2.62	42	*5.33*	*85*
absolute value, z = abs(x)	-	-	1.07 or 1.24	17 or 20	2.14 or 2.50	34 or 40	*5.08*	*81*
Square root, z = sqrt(x)	*52.2*	835	*51.2*	*819*	*52.7*	*843*	*45.8*	*732*
Sine, z = sin(x)	*121.3*	1941	*126.4*	*2023*	*146.2*	*2340*	*123.8*	*1981*
Cosine, z = cos(x)	*120.7*	1931	*130.6*	*2089*	*146.2*	*2339*	*123.1*	*1969*
Tangent, z = tan(x)	*148.1*	2369	*157.1*	*2514*	*176.0*	*2816*	*148.3*	*2373*
Power, z = pow(x,y)	*309.0*	4944	*306.8*	*4908*	*342.9*	*5487*	*301.7*	*4828*
Compare, if(x==y)	0.38	6	0.69	11	1.32	21	*3.95*	*63*
Compare, if (x >y)	0.38	6	0.69	11	1.32	21	*4.10*	*66*
AND, if(x && y)	0.50	8	0.76	12	1.64	26	-	-
OR, if(x || y)	0.50	8	0.76	12	1.64	26	-	-
Bitwise AND, z = x & y	0.44	7	0.88	14	1.75	28	-	-
Bitwise OR, z = x | y	0.44	7	0.88	14	1.75	28	-	-
Bitwise XOR, z = x ^ y	0.44	7	0.88	14	1.75	28	-	-
Bitwise NOT, z = ~x	0.32	5	0.63	10	1.25	20	-	-
Bitshift, z = x << y (see note 3)	*0.69+0.31y*	11+5y	*1+0.31y*	*16+5y*	*1.75+0.44y*	*28+7y*	-	-
conversion to int, z = int(x)	0.44	7	-	-	0.752	12	4.85	78
conversion to long, z =long(x)	0.82	13	1.00	16	-	-	5.17	83
conversion to float, z = float(x)	*3.22*	*52*	*3.89*	*62*	*4.46*	*71*	-	-

Notes:

1. The values reported include the time required to read and write the variables to RAM. The time required by the Arithmetic Logic Unit to perform the elementary operation can be obtained by subtracting the time reported in the first line of the table multiplied by the number of variables involved. For example, the time to add two bytes already in the Register File and store the result back in the Register File is given by : 7 cycles - 2 cycles * 3 variables = 1 cycle.

2. The numbers in grey represent operations for which the calculation time depends on the values of the input variables. The numbers reported represent the average time required to complete the operation (based on at least 100 relevant random numbers).

3. The Bitshift calculation time depends on the number of bit to shift (i.e. the variable y). The Bitshift operation is faster if the value of y is known at compile time.

Calculation speed of an ATmega328-based Arduino running at 16 MHz

(excluding the time required to read and write to RAM - see note 1)

Operations	Byte (1 byte)		Int (2 bytes)		Long (4 bytes)		Float (4 bytes)	
	µs	CPU cycles	µs	CPU cycles	µs	CPU cycles	µs	CPU cycles
Read or write to RAM	**0.13**	**2**	**0.25**	**4**	**0.50**	**8**	**0.50**	**8**
Addition, z = x + y	0.06	1	0.13	2	0.26	4	*6.61*	*106*
Subtraction, z = x - y	0.06	1	0.13	2	0.26	4	*6.65*	*106*
Multiplication, z = x * y	0.25	4	0.75	12	3.03	48	*8.17*	*131*
Division, z = x/y	*12.5*	201	*15.0*	*239*	*39.4*	630	*29.6*	*473*
Modulo, z = x % y	*12.5*	201	*15.0*	*239*	*39.4*	630	*29.6*	*473*
Min and Max, z = min(x,y)	0.35	6	0.54	9	1.12	18	*3.83*	*61*
absolute value, z = abs(x)	-	-	0.56 or 0.75	9 or 12	1.14 or 1.50	18 or 24	*4.08*	*65*
Square root, z = sqrt(x)	*52.0*	831	*50.7*	*811*	*51.7*	*827*	*44.8*	*716*
Sine, z = sin(x)	*121.1*	1937	*125.9*	*2015*	*145.2*	*2324*	*122.8*	*1965*
Cosine, z = cos(x)	*120.4*	1927	*130.1*	*2081*	*145.2*	*2323*	*122.1*	*1953*
Tangent, z = tan(x)	*147.8*	2365	*156.6*	*2506*	*175.0*	*2800*	*147.3*	*2357*
Power, z = pow(x,y)	*308.7*	4938	*306.0*	*4896*	*341.4*	*5463*	*300.2*	*4804*
Compare, if(x==y)	0.13	2	0.19	3	0.32	5	*2.95*	*47*
Compare, if (x >y)	0.13	2	0.19	3	0.32	5	*3.10*	*50*
AND, if(x && y)	0.25	4	0.26	4	0.64	10	-	-
OR, if(x || y)	0.25	4	0.26	4	0.64	10	-	-
Bitwise AND, z = x & y	0.07	1	0.13	2	0.25	4	-	-
Bitwise OR, z = x | y	0.07	1	0.13	2	0.25	4	-	-
Bitwise XOR, z = x ^ y	0.07	1	0.13	2	0.25	4	-	-
Bitwise NOT, z = ~x	0.07	1	0.13	2	0.25	4	-	-
Bitshift, z = x << y (see note 3)	*0.31+0.31y*	5+5y	*0.25+0.31y*	*4+5y*	*0.25+0.44y*	*4+7y*	-	-
conversion to int, z = int(x)	0.07	1	-	-	0.00	0	4.10	66
conversion to long, z =long(x)	0.19	3	0.25	4	-	-	4.17	67
conversion to float, z = float(x)	*2.60*	*42*	*3.14*	*50*	*3.46*	*55*	-	-

Notes:

1. In this table, we assume that all the variables can be processed directly from the Register File of the Arithmetic Logic Unit. The values reported thus exclude the time required to read and write the variables to RAM. The first line of the table can used to calculate the time required to read or write values from/to RAM.

2. The numbers in grey represent operations for which the calculation time depends on the values of the input variables. The numbers reported represent the average time required to complete the operation (based on at least 100 relevant random numbers).

3. The Bitshift calculation time depends on the number of bit to shift (i.e. the variable y). The Bitshift operation is faster if the value of y is known at compile time.

Time required to run various commands on an ATmega328-based Arduino running at 16 MHz

Commands			Duration	
			µs	CPU cycles
Various Arduino commands		Analog read	110.9	1775
		Digital read (pin without PWM)	3.65	58
		Digital read (pin with PWM)	4.53	72
		Digital write (pin without PWM)	3.93	63
		Digital write (pin with PWM)	4.84	77
		Analog write	8.05	129
		millis()	1.32	21
		micros()	2.96	47
		random()	140.3	2245
Assembly commands		Direct read of 8 pins (e.g. PIND)	0.06	1
		Direct read of 8 pins and store result to RAM (e.g. z = PIND)	0.19	3
		Direct write to 8 pins (e.g. PORTD = B00001000)	0.06	1
		Direct write to 8 pins from a byte stored in RAM (e.g. PORTD = z)	0.19	3
		Skip a cycle (__asm__("nop\n\t"))	0.06	1
Serial communication	Buffer free	Serial.write() @ 9600 baud	9.6	154
		Serial.write() @ 115200 baud	10.1	162
		Serial.print("Hello world") @ 9600 baud	113.5	1816
		Serial.print("Hello world") @ 115200 baud	119.0	1904
	Buffer full	Serial.write() @ 9600 baud	1039	16621
		Serial.write() @ 115200 baud	83.6	1338
		Serial.print("Hello world") @ 9600 baud	11439	183021
		Serial.print("Hello world") @ 115200 baud	933.8	14941