

# Math Machines: Designing Motions and More

## MM:DMM Readme (2016)

Learning with Math Machines, Inc.

For HD video introductions to Math Machines, link to [www.mathmachines.net/Video](http://www.mathmachines.net/Video)

## INTRODUCTION

“Math Machines” are simple devices which give an immediate, physical and dynamic expression to “abstract” mathematical equations. Users can enter a virtually unlimited variety of free-form mathematical functions which control motions and colors. Math Machines do not replace paper-and-pencil activities, but instead provide a physical application and immediate, dynamic feedback for those activities. Learners design, test, compare and modify mathematical functions as dynamic relationships. The mathematical level can range from simple proportions and linear functions, through polynomials, exponential and logarithmic functions and trigonometry and to the introduction of differential calculus. Functions can be expressed either explicitly or in recursive form.

This package installs drivers required for both the *Math Machines: Designing Motions and More* software and the associated hardware. The software can be used without external hardware (this is particularly valuable for instructors or students developing or testing functions at home), but full functionality requires a chipKIT WF32 board from Digilent, Inc. and additional hardware. Details are available at <http://www.mathmachines.net>.

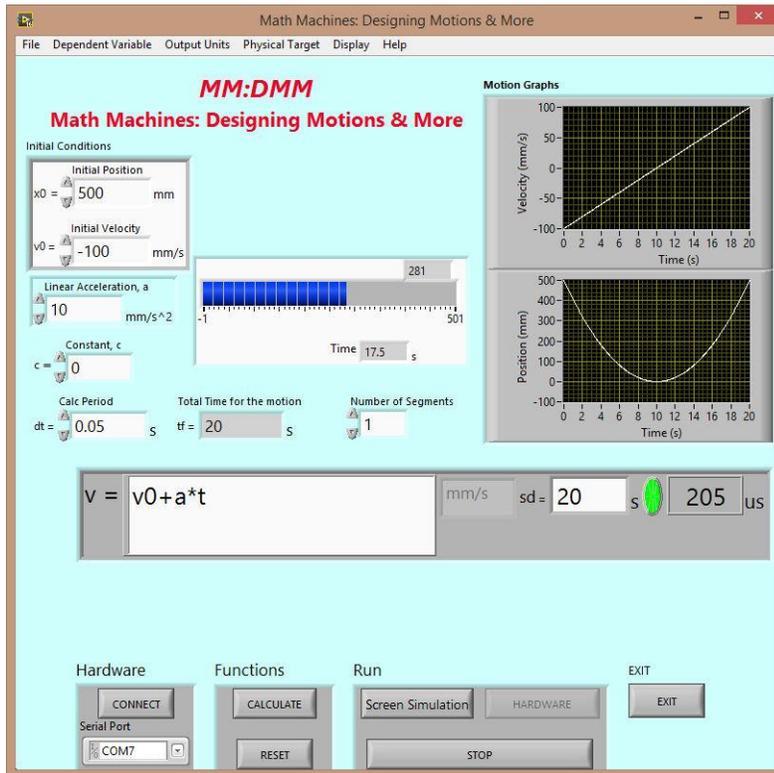
COMPUTER REQUIREMENTS: PC computer with Windows 7 or higher.

### INDEX:

#### Software Options:

Angular or Linear Position, Angular or Linear Velocity	page 2
RGB Colors	page 3
Constant Angular or Linear Velocity	page 4
Earthquake Simulator	page 5
Functions for Controlling Math Machines	page 6
Additional Information	page 7

## Angular Position, Linear Position, Angular Velocity, Linear Velocity



“Angular Position” and “Angular Velocity” modes allow users to input algebraic functions for the position of a servo motor or a stepper motor as a function of time. “Linear Position” and “Linear Velocity” modes are generally used to program position or the velocity of a dot projected from a laser mounted on a servo motor or the motion of a stepper-motor driven cart. The independent variable is time, “t”, the elapsed time in seconds. Outputs are displayed on-screen as numeric values, graphs and dynamically via a dial or slider. If the chipKIT WF32 board and appropriate hardware are attached, the functions control

the physical motion of an actual device.



Optional constants are  $x_0$ ,  $v_0$  and  $a$  (or  $\theta_0$ ,  $\omega_0$  and  $\alpha$ ),  $dt$ , and  $t_f$ . Variables  $x$  and  $v$  (or  $\theta$  and  $\omega$ ) can be used in recursive functions. Functions can be piecewise-defined, optionally using  $t_s$  (elapsed time within the segment) and  $s_d$  (length of the segment). Initial values reset automatically at the beginning of each new segment.

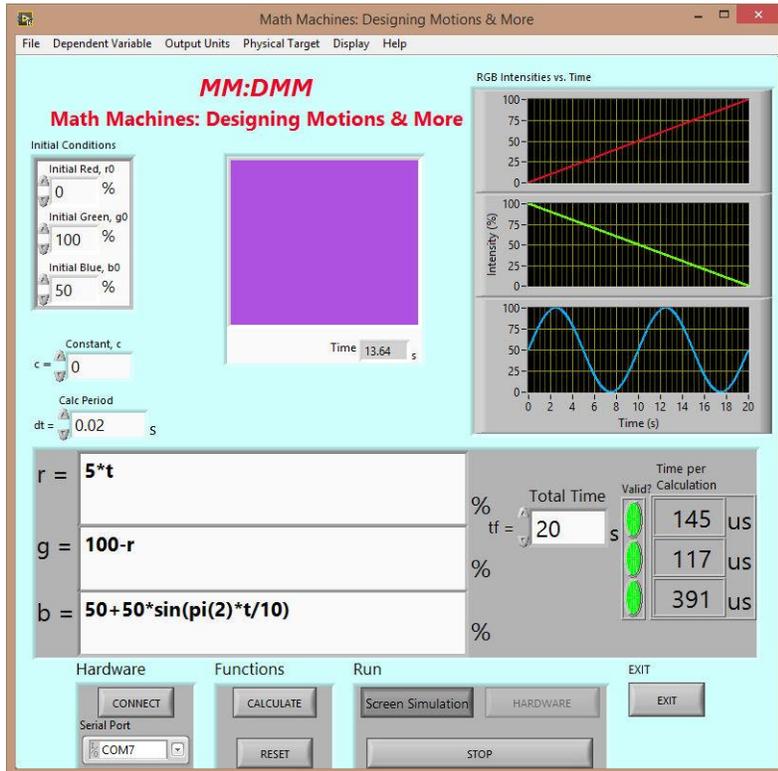
A sample task for learners might be, “Make the laser dot move at constant speed from the frog to the pink target in 20 seconds.”

Assuming learners (or the teacher) have selected millimeters as the units, that the coordinate system follows the markings on a meter stick, and that the objects are located at 300 and 600 mm, solutions could include:

1. “ $x = 300 + 15*t$ , or
2. “ $x = x + 0.75$ ” (with  $x_0 = 300$  and  $dt = 0.05$  s)



## RGB Colors



The RGB Colors mode allows users to enter 3 simultaneous functions, one for each of the 3 primary colors (red, green and blue) used by computer monitors, TVs, billboards and other light-emitting equipment to produce a virtually unlimited range of colors, shades and tints.

The independent variable for each of the functions is "t", the elapsed time in seconds, and the output (dependent variable) of each function is the percentage (from 0 to 100) of its maximum possible intensity. Outputs are displayed on-screen as numeric values, graphs and dynamically via a color box. If

the chipKIT board and Servo/LED shield are attached, the color output can also be displayed on a three-color LED "pixel."

Optionally, a constant, c, can be displayed and used in functions, as can initial values, r<sub>0</sub>, g<sub>0</sub> and b<sub>0</sub>. For recursive functions, r, g and b can also be used as variables, always representing the value calculated during the previous calculation cycle. If desired, dt, dr, dg, db, and tf can also be used as variables.



A sample task for learners might be, "Make the red increase at a steady rate from 0 to 100 in 20 seconds while the green decreases from 100 to 0 at the same time."

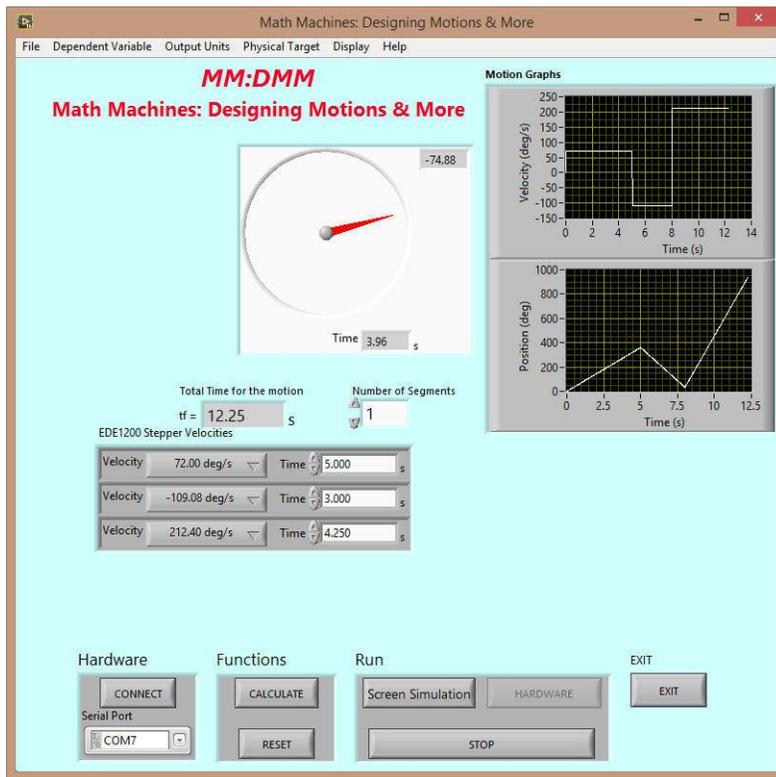
Explicit solutions include:

1. "r = 5\*t", and  
"g = 100 - 5\*t"
2. "r = 5\*t", and  
"g = 100-r"

Recursive solutions include:

3. "r = r+0.1" (with r<sub>0</sub> = 0 and dt=.02), and  
"g = g-0.1" (with g<sub>0</sub> = 100)
4. "r = r+5\*dt" (with r<sub>0</sub> = 0 and with any value of dt), and  
"g = g-5\*dt" (with g<sub>0</sub> =100)

## Constant Angular or Linear Velocity



The Math Machines LACI shield uses an EDE1200 stepper motor controller chip, which can produce continuous motion at a number of pre-set speeds. Using these speeds (rather than algebraic functions) can be useful for some learners, and it can allow us to simplify and shorten programs stored to the chipKIT WF32 board's SD card.

Users select discrete speeds and the duration for each segment and the motion is displayed on-screen to a dial or slider and to graphs of position and velocity. The physical output must be a unipolar stepper motor under control of an EDE 1200 chip.

Linear velocities assume that LACI (or another cart) is being powered via wheels attached to the stepper motor.

Sample tasks using ratios might be:

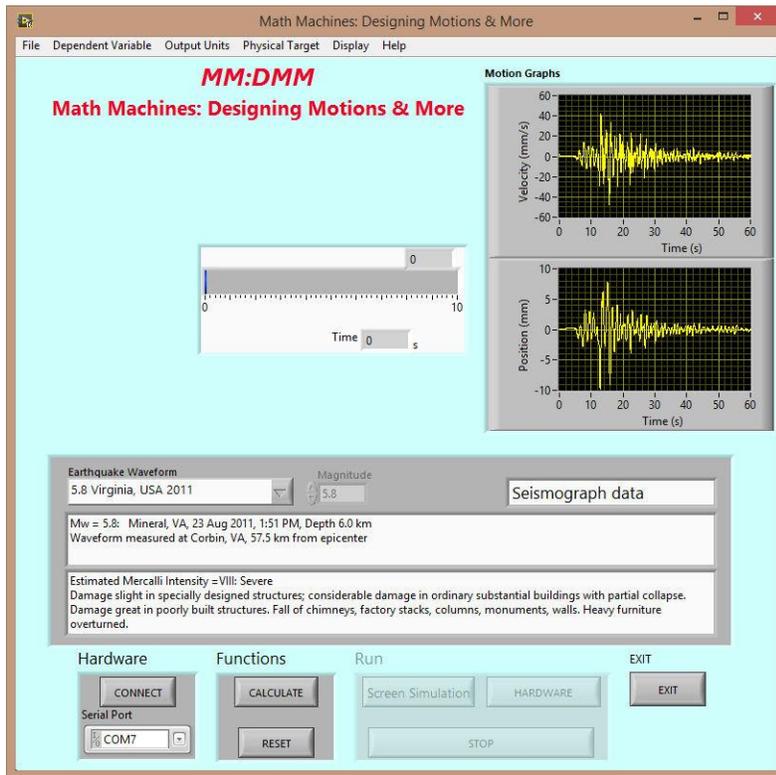
1. Make LACI move 600 mm at a speed of 37.92 mm/s, or
2. Move LACI forward for 10 seconds at a speed of 46.64 mm/s, then return to the starting point at a velocity of -18.88 mm/s.



A more complex task might be:

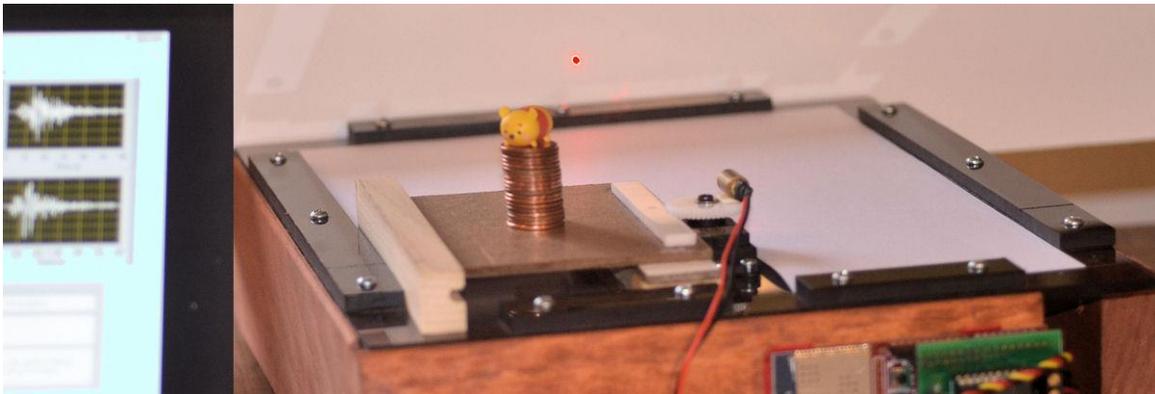
3. Program LACI to "shift up" through several different speeds before reaching its maximum speed of 90.82 mm/s and then to stop suddenly so the passenger and cargo are thrown off at the end of the track.

# Earthquake Simulator



Earthquake mode is designed for use with a small platform attached via a rack and pinion to a servo motor. It allows users to select from a number of actual earthquake wave forms or from a few alternative wave forms (square, sine, triangular) and then to adjust the Richter magnitude of the motion.

While learning about logarithmic scales, users can also discover that Richter magnitude (based on the maximum horizontal displacement) does not tell the entire story about earthquake damage.



Sample tasks include:

Team 1: Without adhesives, build the tallest and least expensive pile of coins you can that will withstand a simulated earthquake with Richter magnitude 4.

Team 2: Design a Richter magnitude 4 earthquake that will damage or destroy Team 1's structure.

# FUNCTIONS FOR CONTROLLING MATH MACHINES

The functions listed below are available for use in the LabVIEW Math Script windows used by Math Machines. Other functions may also be available.

## MOST COMMON FUNCTIONS AND OPERATORS

+, -, *, /	Add, subtract, multiply or divide. Note that <b>multiplication must be explicit</b> . For example “2*t” is valid, but “2t” is not.
^	Power. For example, “5^3” is five cubed.
abs	Absolute value. For example “abs(-7) = 7
pi(1), pi(2), pi(1/2), etc.	Note that “pi” is only allowed when it is followed by a multiplier as shown.

## TRIGONOMETRIC FUNCTIONS

sin, cos, tan, etc.	Trig functions using radians.
asin, acos, etc.	Inverse trig functions using radians
sinh, acosh, etc.	Hyperbolic trig functions

## OTHER FUNCTIONS

sign	Returns -1, 0 or 1. For example, “sign(-8.4)” = -1.
rand	Returns a random number between 0 and 1. Requires a “seed” such as rand(5) or rand(t).
sqrt	Square root function. For example “sqrt(4)” = 2
log	Base 10 logarithm
log2	Base 2 logarithm
ln	Natural logarithm
exp	Exponential function. For example, “exp(3)” = e^3 where e = 2.71828.... (Inverse of ln)
ceil	Rounds towards + infinity. For example, “ceil(2.3)” = 3
floor	Rounds towards – infinity. For example, “int(4.6)’ = 4 and “int(-5.2)’ = -6.
int	Same as “floor”
step	For example, “step(t-5)” has the value 0 when t<5 and 1 when t>5.
spike	For example, “spike(t-5)” has the value 1 when 5<t<6 and zero at all other times.
square	For example, “square(t)” alternates each second between 0 and 1

# SOFTWARE ADDITIONAL INFORMATION

The software is licensed for educational use under terms of the Math Machines Software License Agreement.

Learning with Math Machines, Inc. is a non-profit, 501(c)(3) organization. If you find this software useful, please consider contributing \$25 (or whatever you can afford) through our website, [www.mathmachines.net](http://www.mathmachines.net).

This material is based in part upon work supported by the National Science Foundation' Advanced Technological Education Program under Grants No. DUE-0202202 and DUE-1003381. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Please email [fred.thomas@mathmachines.net](mailto:fred.thomas@mathmachines.net) with feedback, to learn about opportunities for collaboration or to inquire about the possibility of a Math Machines workshop in your area.