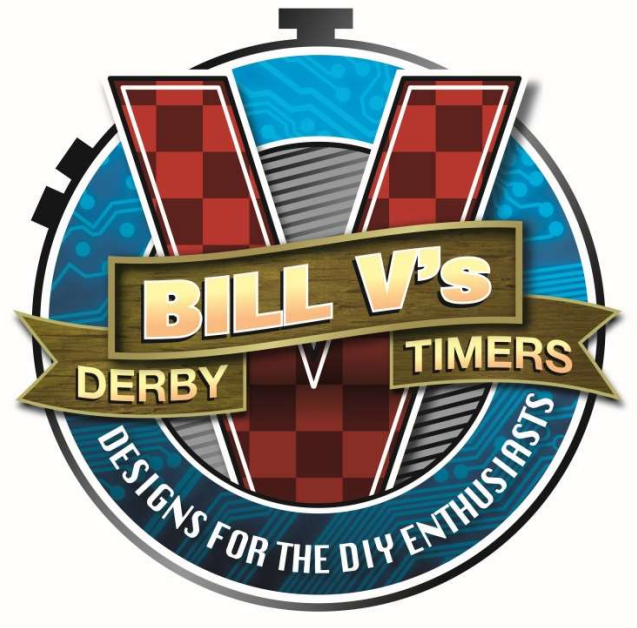
BILL V'S DERBY TIMERS

Arduino Based Pinewood Derby 3-Lane Racetrack Timer

Design Version 1a
For Use with Pinewood Master™ Race Management Software by Marc Speth
(Apple MAC OS-X)

USER MANUAL



1 INTRODUCTION

The Pinewood Derby racetrack timer discussed herein is a relatively simple, yet practical design that provides the features needed to conduct a successful race event. This version is specifically designed to operate with the Apple Macintosh based Pinewood Master™ race management software by Marc Speth (http://www.pinewoodmaster.com/index.html) to control the race timer and display the race results. This is accomplished by emulating the Micro Wizard Fast Track™ timer serial communications protocol that the Pinewood Master™ race management software supports.

The Arduino UNO™ board provides the interfaces to the optical lane sensors, race start (gate) switch, reset switch and 3 race status LEDs. It executes the software that performs the timing, monitors the track status and sends the race results to the Mac for post-race processing and display by the Pinewood Master software. Data is transferred between the Arduino timer and the PC via the USB interface.

Features included in this version of the Arduino based timer software include:

- Timing accuracy rounded to the nearest millisecond (To support Pinewood Master format).
- Lane masking (aka: "BYE" lane) capability (*Not supported by Pinewood Master™ race management software*).
- Automatic timer reset when gate switch returned to the closed position.
- Timer held in the "Not Ready" state when a lane sensor is obstructed or the gate switch is in the wrong state to start the race.

Note: *The Pinewood Master™ race management software is not provided as part of this package and must be purchased by the end user at: http://www.pinewoodmaster.com/index.html.*

2 SOFTWARE INSTALLATION

NOTE: The following instructions are for the installation of the customized Pinewood Derby Elapsed Time Timer software consisting of the source code file PWD_Mac_Timer03_Ver1a.ino.

2.1 Arduino UNO Software Installation

2.1.1 Arduino Software Environment Installation – MAC OS X

Download and install the Arduino Integrated Development Environment (IDE). For complete step-by-step instructions on how to download and install the Arduino Integrated Development Environment (IDE) on the Mac OS X, visit https://www.arduino.cc/en/Guide/MacOSX. The WEB site also provides tutorials to help you every step of the way.

2.1.2 Arduino Source Code Installation

Once you have the Arduino development environment installed (per Section 2.1.1) perform the following to install the Arduino source code:

1. Extract the file 'PWD_Mac_Timer03_Ver1a.ino' from the zip file provided and copy it into the Arduino Sketchbook folder.

2.1.3 Uploading the PWD_Mac_Timer code to your Arduino Board

Perform the following steps to upload the Timer code to the timer's Arduino board:

Step	Action	Comments
1	Ensure your MAC PC is powered up and ready.	
2	Connect the track timer (Arduino board) to your USB port via a USB cable.	Verify the Arduino board is powered up (Power LED is illuminated)
3	Double click on the 'PWD_Mac_Timer03_Ver1a.ino' file to launch the file and bring up the Arduino Integrated Development Environment (IDE).	Arduino IDE window is displayed with the PWD_Mac_Timer03_Ver1a source code listing.
4	Select the 'Tools/Board' pull-down menu to select/verify the Arduino Uno board is selected.	
5	Select the 'Tools/Port' pull-down menu to select/verify the COM port selection (i.e. COM1, COM2, etc.).	Arduino COM port is selected / verified.
6	Select "→" (upload) to start the compile and upload process.	The program will compile and automatically upload to the Arduino board.
7	**OPTIONAL** To verify a successful upload, perform the diagnostic test procedure located at the end of this document.	

Refer to the Arduino website for more in depth instructions if problems are encountered. Note that once the software has successfully been uploaded to the Arduino board, it is there permanently, unless overwritten by another upload. Hence, you only have to perform these steps once. Future use of the Arduino timer during your race events does not require an upload. Just connect the Arduino timer to your PC, launch the Display software and you should be ready to go.

2.2 Pinewood Master Race Management Software Installation

Please refer to the Pinewood Master™ documentation and website (www.pinewoodmaster.com) for installation of the Pinewood Master race management software.

NOTE: *Your MAC may require a serial COM port driver update in order for the Pinewood Master software to communicate with the Arduino based timer. Please visit http://plugable.com/prolific for download and installation of the PL2303 driver specific to your MAC version. You can also get more info from the Pinewood Master web site at: http://www.pinewoodmaster.com/fast-track.html .*

2.2.1 Pinewood Master Com Port Settings

In order for Pinewood Master race management software to communicate with the Arduino timer make the following settings in the Pinewood Master software (See Fig. 1):

- Timer: Fast Track
- Baud Rate: 9600
- Data Bits: 8
- Stop Bits: 1
- Parity: None

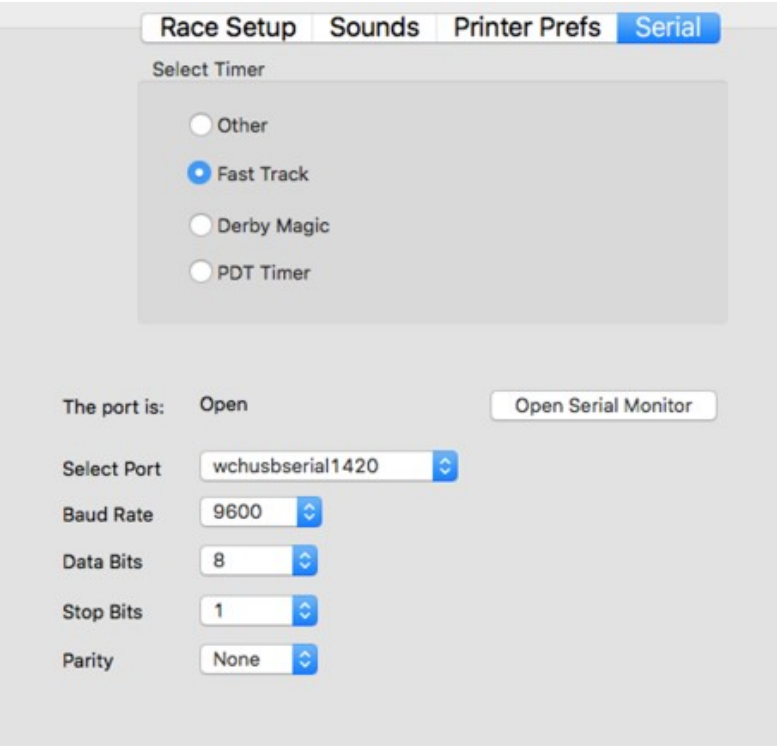


Figure 1. Example Serial Port Setup

If successful, running a test race while observing the Serial Monitor will yield results similar to those shown in Figure 2 below except lanes 4, 5 & 6 will show 0.000.

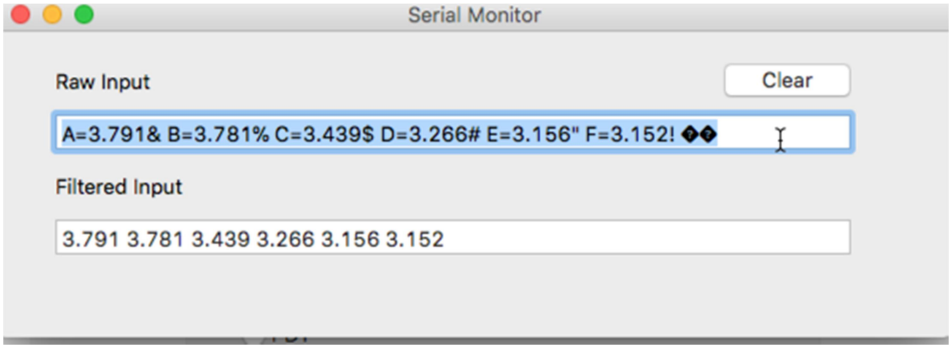


Figure 2. Example Serial Monitor Results

3 TIMER OPERATION

3.1 Arduino Timer Operation Overview

Operation of the Arduino based timer is straight forward. The track timer code running on the Arduino cycles through four states as follows:

- Track Not Ready
- Ready
- Racing
- Finished

The "Track Not Ready" state is entered at initial power-up. The "Track Not Ready" state is also entered when the Arduino timer senses that one or more of the optical lane sensors is obstructed or the Gate Switch is in the wrong state to start the race. Either condition will cause the timer to remain in the "Not Ready" state until the issue is resolved. Illumination of the red LED indicates the timer is in the Not Ready state.

The "Ready" state is entered when the timer detects that the optical lane sensors and Gate Switch are in the correct state to start the race. Illumination of the yellow LED indicates the timer is in the Ready state.

The "Racing" state is displayed when the timer has sensed activation of the Gate Switch indicating the cars have left the gate and the race is underway. Illumination of the green LED indicates the timer is in the Racing state.

The "Finished" state is entered when all cars have crossed the finish line or when 10 seconds have elapsed, whichever occurs first. Additionally, the "Finished" state is entered if the Gate Switch is returned to the closed position prior to the10-second time-out. The 10-second time-out is for cases where a car fails to cross the finish line or a lane was not used. The 10-second time-out does not apply for lanes that have been masked (BYE lane) and thus ignored by the timer. When in the timer transitions from the "Racing" state to the "Finished" state, race results are transmitted to the race management software via the serial COM port. Then, after a short delay, the timer is reset back to the Not Ready state until it senses the track is ready to start the next race.

3.2 Running a Race Using the Race Management Software

Perform the following steps to run a race using the race management software:

1. Ensure the timer is powered up and connected to the MAC PC.
2. Start the race management software and confirm the COM port selection.
3. Ensure the gate switch is closed and all lane sensors are un-obstructed
    a. The yellow 'Ready' LED indicator is illuminated.
4. Stage the cars on the track at the start line.
5. When ready, activate the Gate Release handle/mechanism. *Note: Do not return the start gate to the closed position until the race has completed. Doing so will reset the timer*.
    a. The green 'Racing' LED indicator is illuminated.
6. Wait for all cars to cross the finish line or when 10 seconds has elapsed, whichever occurs first.
    a. Race results are displayed.
    b. The red 'Not Ready' LED indicator is illuminated
7. Return the Gate Release handle/mechanism to the closed position
    a. The yellow 'Ready' LED indicator is illuminated.
8. Repeat steps 4 through 7 till all heats have been ran.

4 Arduino Timer Communication

Communication between the Arduino based timer and the MAC PC is via the USB interface which has been set up as a serial link running at 9600 baud, 8 bits, no parity, and 1 stop bit (9600/8-N-1). ASCII character strings transmitted between the timer and the MAC PC are used to control the timer as described in the table below.

***Note**: As of this writing, it has been confirmed that the Pinewood Master Race management software does not send any data to the timer. It only receives the messages from the timer and interprets them.*

ASCII Command	Direction	Description
R	PC-to-Arduino	Reset – Resets the Arduino timer. *(Not supported by Pinewood Master)*
MA	PC-to-Arduino	Lane 1 Mask command. Instructs Arduino timer to set the Lane 1 mask flag to cause the timer to ignore that lane. *(Not supported by Pinewood Master)*
MB	PC-to-Arduino	Lane 2 Mask command. Instructs Arduino timer to set the Lane 2 mask flag to cause the timer to ignore that lane. *(Not supported by Pinewood Master)*
MC	PC-to-Arduino	Lane 3 Mask command. Instructs Arduino timer to set the Lane 3 mask flag to cause the timer to ignore that lane. *(Not supported by Pinewood Master)*
MG	PC-to-Arduino	Un-mask command. Instructs Arduino timer to clear all mask flags. *(Not supported by Pinewood Master)*
@	Arduino-to-PC	Gate Status – Informs the race management software that the Gate Switch transitioned from open to closed. *(Not supported by Pinewood Master)*
*	Arduino-to-PC	Mask command acknowledgement character

In addition to the above serial messages the Arduino passes the race results to the PC display software via a single ASCII string having the following format:

A=N.NNNX B=N.NNNX C=N.NNNX D=N.NNNX E=N.NNNX F=N.NNNX<cr><lf>

Where: A - F reference lanes 1 - 6 respectively
N.NNN is the lane time in seconds
X is the finish sequence character as follows:

! = 1'st place	" = 2'nd place
# = 3'd place	$ = 4'th place
% = 5'th place	& = 6'th place

Troubleshooting: Use of the Arduino IDE serial monitor or another serial terminal program can be used to observe these commands. Refer to the diagnostic test procedure on the following pages to assist in troubleshooting any issues. Feel free to contact the author via email at billv923@outlook.com for additional help.

DIAGNOSTIC TEST PROCEDURE

This test procedure was written to assist in testing and troubleshooting the Arduino based timer hardware and software.

Setup:

- Bring up the Arduino Integrated Development Environment (IDE) on your PC (Refer to Section 2.1)
- (Optional) Select and load the PWD_Mac_Timer03_Ver1a.ino file.
- Select the 'Tools/Port' pull-down menu to select/verify the port selection (i.e. COM1, COM2, etc.).
- Open the serial monitor by clicking on the little magnifying glass near the upper right corner of the IDE display. Ensure the baud rate is set to 9600.

Perform the following test procedure.

Step	Action	Expected Results
1	Ensure the Gate Switch is closed and the optical lane sensors are properly illuminated (not obstructed).	N/A
2	On the serial monitor enter the letter R in the command line, press ENTER or click on the Send button.	• The yellow LED is illuminated. If the red 'Not Ready' LED remains illuminated the Gate Switch in in the wrong state or one or more optical lane sensors are not properly illuminated (obstructed?). Troubleshoot and correct the problem before continuing.
3	Obstruct (block) the light illuminating the Lane 1 optical sensor. Then on the serial monitor enter the letter R in the command line, press ENTER or click on the Send button.	N/A • The yellow 'Ready' LED extinguishes. • The red 'Not Ready' LED illuminates.
4	Remove the obstruction from the Lane 1 optical sensor.	• The red 'Not Ready' LED extinguishes. • The yellow 'Ready' LED illuminates.
5	Repeat steps 3 & 4 for each lane.	Same as steps 3 & 4.
6	Set the Gate Switch to the open position.	• The yellow 'Ready' LED extinguishes. • The green 'Racing' LED illuminates.
7	Wait 10 seconds.	After 10 seconds: • The red 'Not Ready' LED illuminates. • The following text string is displayed on the monitor: "A=9.999 B=9.999 C=9.999 D=0.000 E=0.000 F=0.000"
8	Ensure the optical lane sensors are properly illuminated (not obstructed), then Close the Gate Switch.	• The '@' symbol is displayed on the monitor. • The red 'Not Ready' LED extinguishes. • The yellow 'Ready' LED illuminates.

Step	Action	Expected Results
9	Set the Gate Switch to the open position.	• The yellow 'Ready' LED extinguishes. • The green 'Racing' LED illuminates.
10	Prior to 10 seconds elapsing wave your hand from left-to-right to sequentially block the light from lane 1 through 3 respectively.	• The following text string is displayed on the monitor: A=X.XXX! B=X.XXX" C=X.XXX# D=0.000 E=0.000 F=0.000 ○ Where X.XXX is a time value and the character just to the right of the time value is the finish order character in accordance with time string format described earlier. • The red 'Not Ready' LED illuminates.
11	Repeat steps 8 thru 10 as desired obstructing the light to the lane sensors in different sequences.	Same results as steps 8 thru 10 except the lane times are different and finish order characters vary in accordance with the sequence the sensors were tripped.
12	Ensure the optical lane sensors are properly illuminated (not obstructed), then Close the Gate Switch.	• The red 'Not Ready' LED extinguishes. • The yellow 'Ready' LED illuminates.
--	TEST COMPLETE	