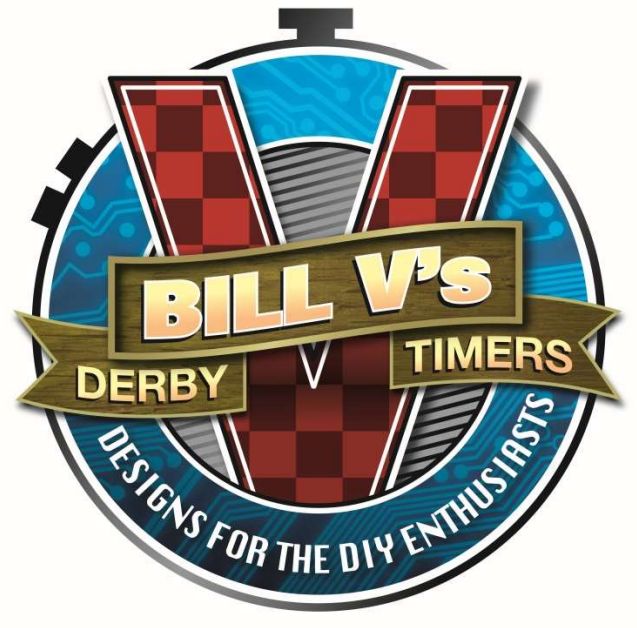
BILL V'S DERBY TIMERS

Arduino Based Pinewood Derby 3-Lane Racetrack Timer

With GrandPrix™ Race Manager (GPRM) Interface

USER MANUAL



1 INTRODUCTION

The Pinewood Derby racetrack timer discussed herein is a practical “minimal hardware” design that provides the features needed to conduct a successful race event. This version is specifically designed to operate with the GrandPrix Race Management™ (GPRM) software by Lisano Enterprises running on a PC to control the race timer and display the race results.

The Arduino UNO™ board used in this project provides the interfaces to the optical lane sensors, race start (gate) switch and executes the software that performs the timing, monitors the track status and sends the race results and track status to the PC for post-race processing and display by the GPRM software. Data is transferred between the Arduino and the PC via the USB interface.

1.1 GrandPrix Race Manager™ Software Package

The commercially available GrandPrix Race Manager™ software by Lisano Enterprises not only provides for control of the track timer but also integrates many features that allow you to manage all aspects of the race event including registering racers and cars, selection of scoring methods, generating schedules, automatic recording of race results, determining awards, providing reports, and more. It is highly customizable making your race more organized and more enjoyable for all.

> ***Note:*** *The GPRM software is not provided as part of this package and must be purchased by the end user. Refer to the web site http://grandprix-software-central.com/ for purchase details and operating instructions.*

2 SOFTWARE INSTALLATION

NOTE: The following instructions are for the installation of the customized Pinewood Derby GPRM Timer software consisting of source code files PWD_GPRM_Timer03_Ver3.ino.

2.1 Arduino UNO Software Installation

2.1.1 Arduino Software Environment Installation

Download and install the Arduino Integrated Development Environment (IDE) onto your PC. For complete step-by-step instructions on how to download and install the Arduino Integrated Development Environment (IDE), visit https://www.arduino.cc/ . Versions are available for Windows, Mac OS X, and Linux. The environment makes it easy to write code and upload it to the board. The WEB site also provides tutorials to help you every step of the way.

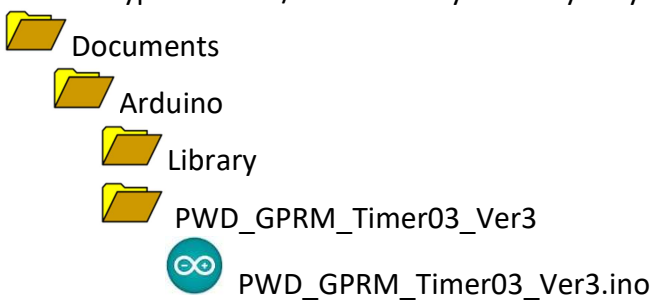
***NOTE**: Installation of the Arduino IDE also installs the necessary USB drivers for your PC to communicate with the Arduino via a serial RS-232 type (i.e. COM1, COM2, etc.) communication link. These drivers are also used by the GPRM software discussed in Section 2.2 below.*

2.1.2 Arduino Source Code Installation

Once you have the Arduino development environment installed on your PC (per Section 2.1.1) perform the following to install the Arduino source code. Note that the Arduino source code file must reside in a folder having the same name (less the '.ino' file extension) in order to launch correctly.

1. Create a subfolder called 'PWD_GPRM_Timer03_Ver3' under the "Arduino" folder. The "Arduino" folder was created during the install, usually under your Documents folder.
2. Extract the file 'PWD_GPRM_Timer03_Ver3.ino' from the zip file provided and copy it into the 'PWD_GPRM_Timer03_Ver3' folder created in the previous step.

This is a typical folder/file hierarchy but may vary:



2.1.3 Uploading the GPRM Timer code to your Arduino Board

Perform the following steps to upload the PWD GPRM Timer code to the timer unit's Arduino microcontroller. 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 GPRM software and you should be ready to go.

Step	Action	Comments
1	Ensure your PC is powered up and ready.	
2	Connect the track timer (Arduino board) to your PC USB port via the USB cable.	Arduino board power indicator illuminates.
3	Double click on the 'PWD_GPRM_Timer03_Ver3.ino' file to launch the file and bring up the Arduino integrated development environment.	Arduino integrated development environment window is displayed with the PWD_GPRM_Timer03_Ver3 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.	

2.2 GrandPrix Race Management™ Setup

This version of the Arduino timer software has been specifically coded to work with the GranPrix Race Management™ (GPRM) software running on your PC. The GPRM software must be custom configured to operate with the Arduino based timer. Single letter commands are used to pass track status and control messages between the timer and the GPRM software. Table 1 below summarizes these commands.

Table 1. GPRM Timer Control Codes

Char.	Source	Description
G	From GPRM	Check Gate request – Sent by GPRM to determine if the timer is ready (See 'O' command)
R	From GPRM	Reset timer request – Sent by GPRM to issue a timer reset
M	From GPRM	Lane mask request – Sent by GPRM to mask a given lane. This command is followed by the lane number (i.e. 'M2' will request the timer mask lane 2)
U	From GPRM	Lane Unmask request – Sent by GPRM to unmask all lanes.
O	To GPRM	Gate Open – Response to Check Gate request if the Gate switch is open (Not Ready)
K	To GPRM	Timer Ready – Response to a Reset request if the timer is in the 'Ready' state
B	To GPRM	Timer Start – Sent when timer transitions from Ready to Racing (Gate switch released)

Refer to the GPRM user manuals and guides for operating the timer using the GPRM software.

2.2.1 GPRM Software Options Setup

***Note**: The example GPRM screen shots used herein are from version 15 of the GPRM software. Earlier or later versions may have slight variations.*

Refer to Figures 1, 2, and 3 below for a typical software setup. Also refer to the GPRM user's manual and help guides as required. Settings listed below with an asterisk (*) are mandatory for this custom timer.

On the Software Settings – General Tab window (Fig. 1):

- Select 'Race car' for the race vehicle.
- Select the 'Pro' User Mode.*
- Set the number of lanes to 3 for the Track setting.*
- Select numbers for the lane labels.

On the Software Settings – Racing Tab window (Fig. 2):

- Ensure you set the track length (start line to finish line).*

On the Software Settings – Standings Tab window (Fig. 3):

- Set options as desired.

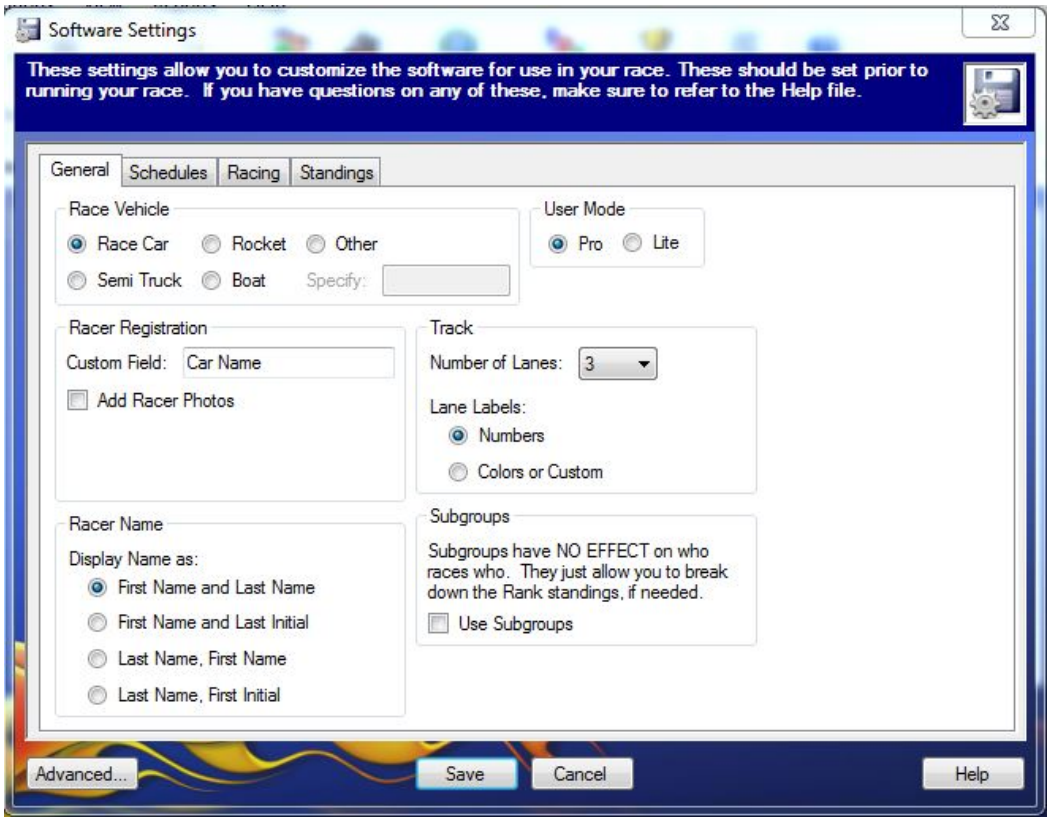


Figure 1. GPRM Software Setup Selections – General Tab

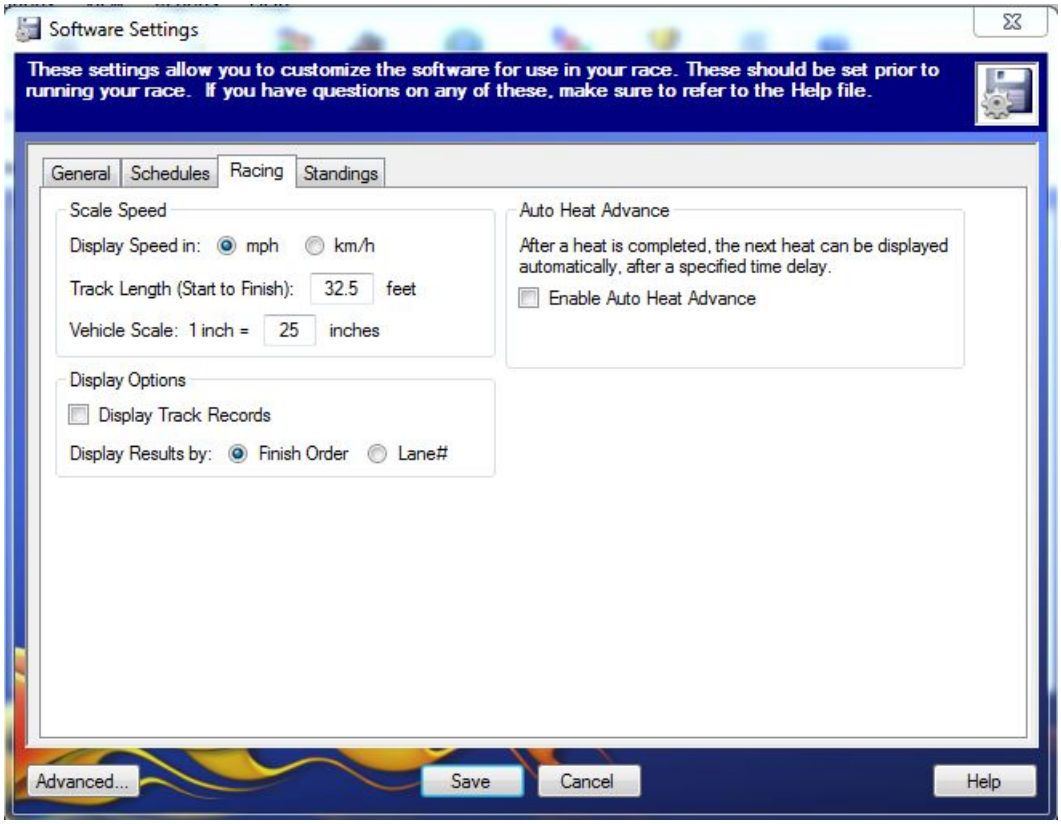


Figure 2. GPRM Software Setup Selections – Racing Tab

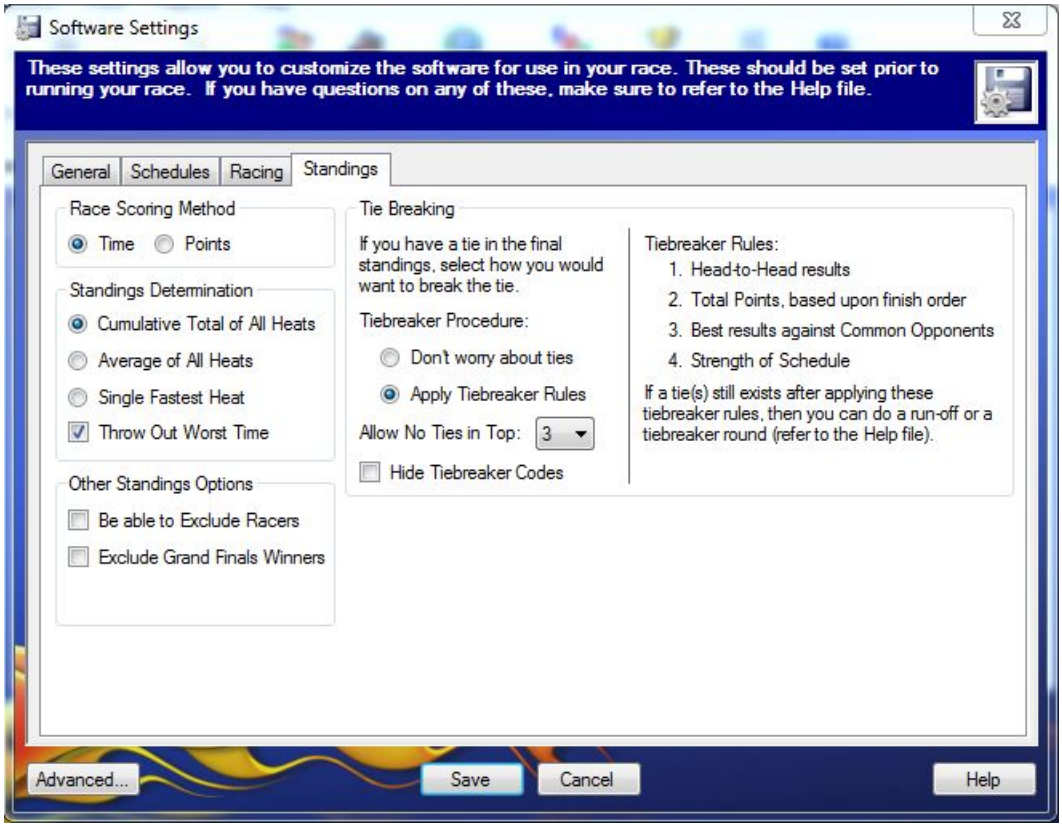


Figure 3. GPRM Software Setup Selections – Standings Tab

2.2.2 GPRM Hardware Setup

***Note**: The example GPRM screen shots used herein are from version 15 of the GPRM software. Earlier or later versions may have slight variations.*

Refer to Figures 4, and 5 below for a typical hardware setup. Also refer to the GPRM user's manual and help guides as required. Settings listed below with an asterisk (*) are mandatory for this custom timer.

On the Hardware Settings – Timer Tab window (Fig. 4):

- Select 'Other' for the Timing System.*
- Set the 'Did Not Finish (DNF) Time' to 9.9999.*
- Put a check in the 'In Use' check-box for each lane you plan on using during your race event.* Note that if you will be generating "PPN" type racing charts up to 6 lanes can be selected. Unused lanes will be marked as "BYE" lanes when the race schedule is generated.
- Verify/select the correct COM port (timer must be connected & powered up)*.

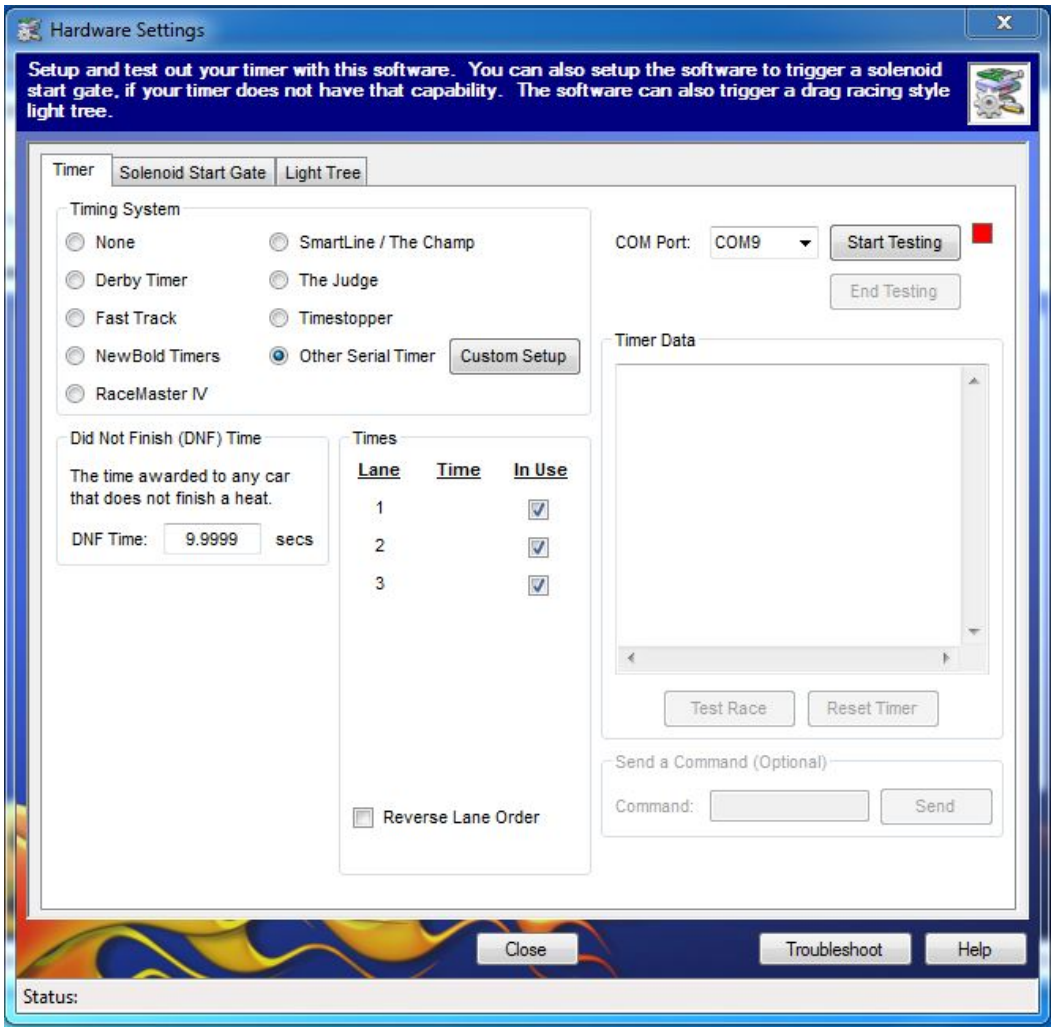


Figure 4. GPRM Hardware Setup Selections – Timer Tab

Click on the "*Custom Setup*" button. On the resulting 'Custom Serial Timer Settings' window (Fig. 5) enter the following configuration settings, and then click SAVE at the bottom of the window.

- COM Port Settings
  - Baud: 9600
  - DataBits: 8
  - Parity: None
  - Stop Bits: 1
- Start Gate message Codes
  - Check Command: G
  - Open Response: O
  - Timer Start Message: B
- Timer Reset message codes
  - Reset Command: R
  - Ready Response: K
  - Response Delay: 0.25 secs

- Lane Masking message codes
  - Mask Command: M
  - Remove all Masks: U
- Other Software Commands
  - Retrieve Data: (leave blank - not used)
  - Force Data Send: (leave blank - not used)
  - Trigger Solenoid: (leave blank - not used)
- Miscellaneous*
  - Read Delay: 0.25 secs
  - Precision: 0.0001 secs
  - Max Time: 9.9999 secs
- Lane Labels:
  - Numbers (1, 2, 3, etc.) selected

* **Note**: The Precision and Max Time normally default to 3 or 4 digits to the right of the decimal point (i.e. 0.0001 & 9.9999 seconds respectively). Make sure they are set as shown to 4 digits to the right of the decimal point or you will encounter problems.

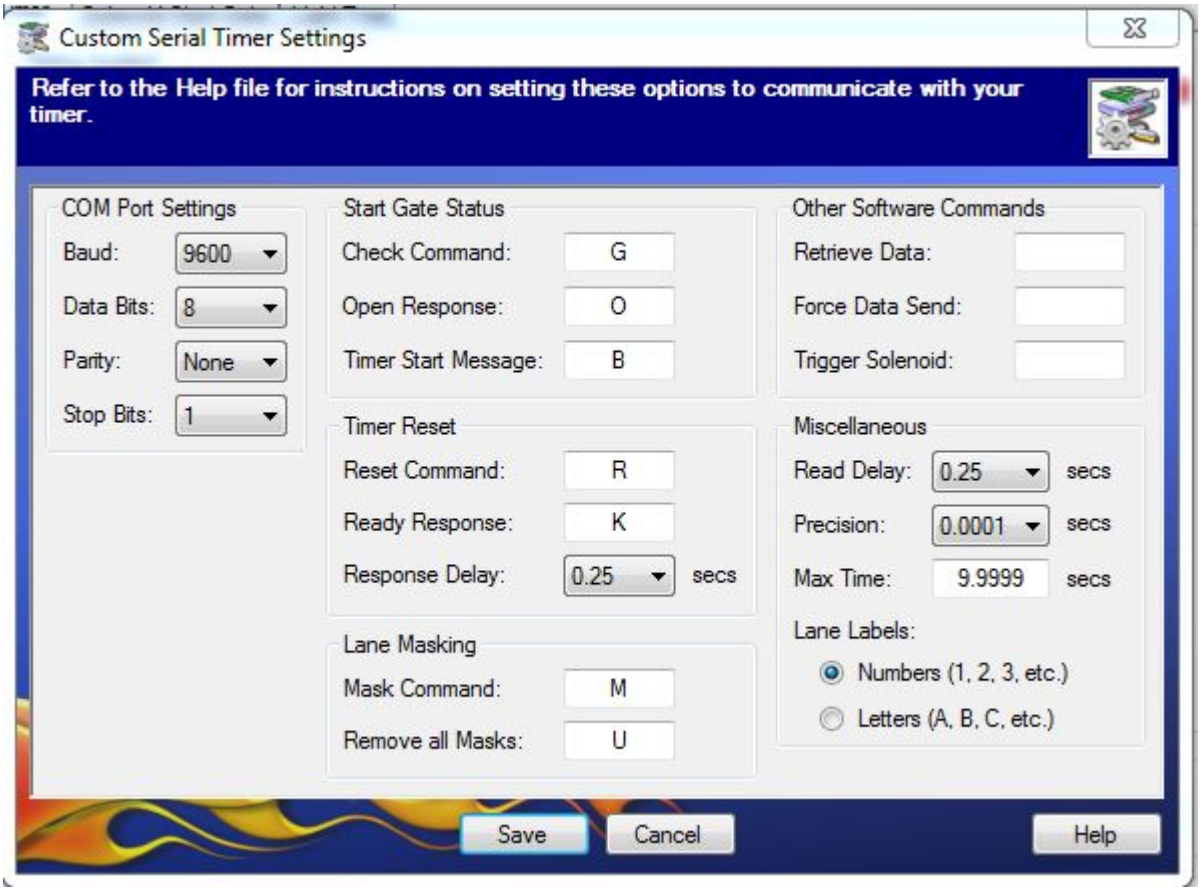


Figure 5. GPRM Hardware Setup Selections – Custom Setup

3 TIMER OPERATION DESCRIBED

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 receives a reset command and it 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. The “Track Not Ready” state is indicated on the GPRM software as a red status box indication on the Racing Page (lower left) with a pop-up “Timer Not Ready” message.

The “Ready” state is entered when the timer receives a reset command and the optical lane sensors and Gate Switch are in the correct state to start the race. The “Ready” state is indicated on the GPRM software as a yellow “Ready” status box indication on the Racing Page (lower left).

The “Racing” state is entered when the timer has sensed activation of the Gate Switch indicating the cars have left the gate and the race is underway. The “Racing” state is indicated on the GPRM software as a green “Start” status box indication on the Racing Page (lower left).

The “Finished” state is entered when all cars have crossed the finish line or when 10 seconds have elapsed, whichever occurs first. 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 designated as BYE lanes and thus ignored by the timer.

3.2 Running The Race

Refer to the GPRM user manuals and help guides for operating the timer using the GPRM software.

The following is a typical step-by-step procedure for running a race. It is assumed the user has already created the race roster and generated a race schedule in GPRM and is ready to begin racing.

Step	Action	Expected Results
1	With the ‘Racing’ screen selected (See Fig. 6), click on the “Ready Timer” button located in the lower left section of the screen.	Button legend changes to “Ready” and the surrounding area is highlighted in yellow as shown in Figure 7. The timer is now ready to race. The yellow highlight indicates that the timer has reported to GPRM that it is ready.

Step	Action	Expected Results
2	With the cars staged on the track, release the start gate to start the race	The “Ready” button located in the lower left section of the screen changes to “Start” and the area is highlighted in green as shown in Figure 8. The green highlight indicates that the timer has reported to GPRM that the gate switch has been activated and the race is underway.
3	Wait for the race to complete.	Race results are displayed on the GPRM ‘Racing’ screen. The green highlighted area is now back to gray. See Figure 9.
4	Click on the “Next” button located in the lower left section of the screen.	The screen advances to the next heat.
5	Repeat steps 1 thru 4 for all heats.	

Figure 6. Ready Timer Racing Screen - Example

Figure 7. Ready Racing Screen - Example

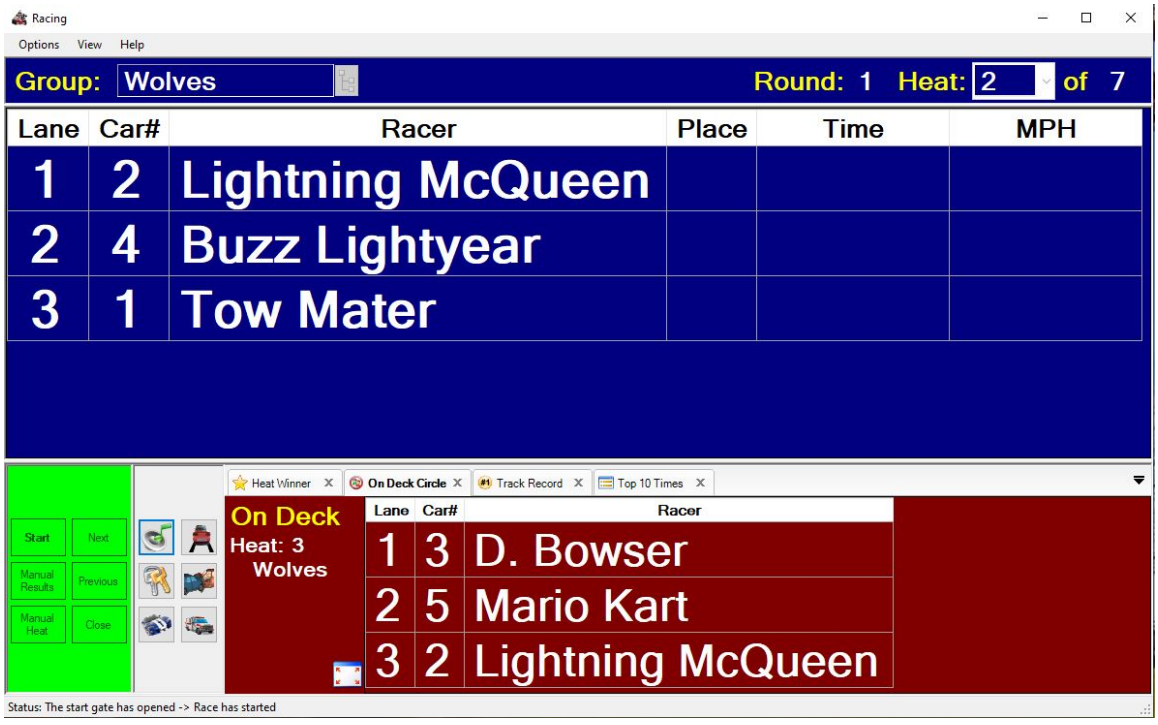


Figure 8. Start (Race in Progress) Racing Screen - Example

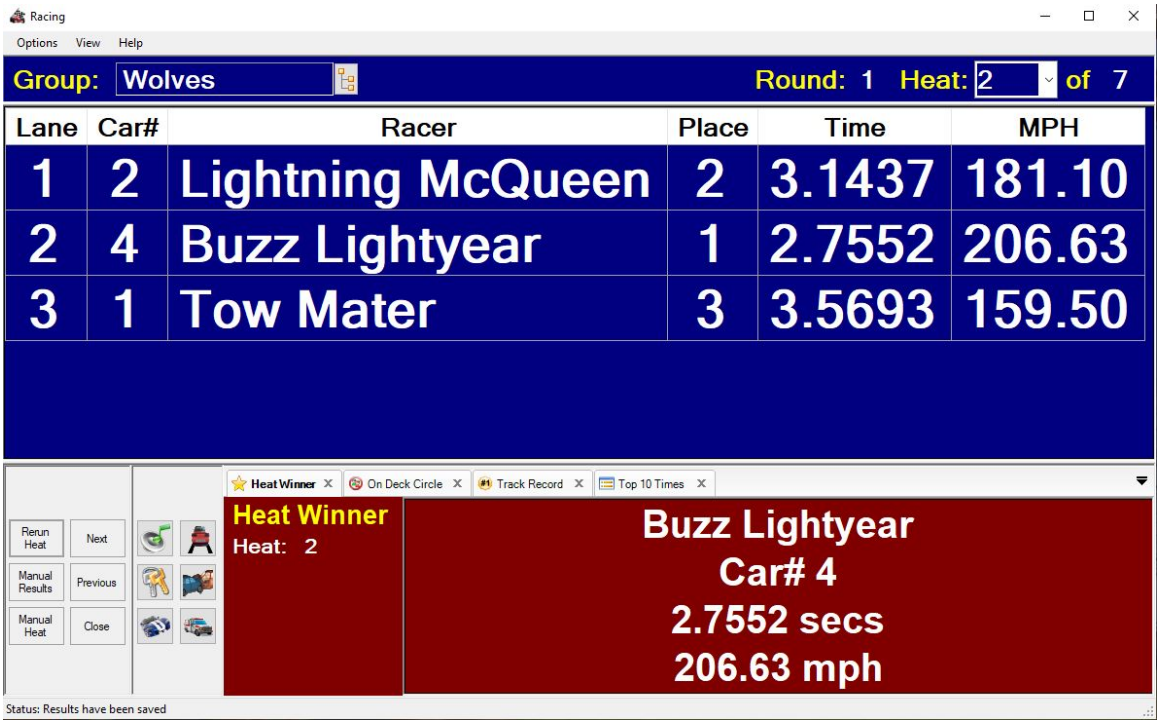


Figure 9. Race Finished Screen - Example

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_GPRM_Timer03_Ver3.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 D in the command line, press ENTER or click on the Send button.	The following message is displayed on the monitor:
		`Debug Status:`
		`- All Inputs OK`
		
		If the gate switch, reset switch or a lane sensor is found to be in the wrong state the debug status message will contain one or more of the following:
		`- Gate Switch: OPEN`
		`- Reset Switch: CLOSED`
		`- Lane 1 OBSTRUCTED`
		`- Lane 2 OBSTRUCTED`
		`- Lane 3 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 letter "O" is displayed on the monitor indicating the track is not ready.
4	Repeat step 3 for each lane.	Same as step 3.
5	Set the Gate Switch to the open position. On the serial monitor enter the letter R in the command line, press ENTER or click on the Send button.	N/A The letter "O" is displayed on the monitor indicating the track is not ready.

Step	Action	Expected Results
6	Set the Gate Switch back to the closed position.	N/A
7	Ensure the Gate Switch is closed and the optical lane sensors are properly illuminated.	N/A
8	On the serial monitor enter the letter R in the command line, press ENTER or click on the Send button.	The letter “K” is displayed on the monitor.
9	Momentarily toggle the Gate Switch to the open position and then back to the closed position.	The letter “B” is displayed on the monitor indicating the race is in progress.
10	Wait 10 seconds.	After 10 seconds: • The following time message is displayed: 1 9.9999 2 9.9999 3 9.9999
11	Ensure the Gate Switch is closed and the optical lane sensors are properly illuminated.	N/A
12	Momentarily depress the RESET switch.	The letter “K” is displayed on the monitor.
13	Momentarily toggle the Gate Switch to the open position and then back to the closed position.	The letter “B” is displayed on the monitor indicating the race is in progress.
14	Wave your hand over all lane sensors to obstruct the light illuminating them before the 10 second timeout has elapsed.	The moment the last lane sensor is tripped: • Times are displayed on the monitor in the following format: 1 x.xxxx 2 x.xxxx 3 x.xxxx Where x.xxxx is the time in seconds.
15	Repeat steps 11 thru 14 as desired obstructing the light to the lane sensors in different sequences.	Same results as steps 11 thru 14.
16	On the serial monitor enter the letter R in the command line, press ENTER or click on the Send button.	The letter “K” is displayed on the monitor.
--	TEST COMPLETE	

Troubleshooting: Use of the Arduino IDE serial monitor or another serial terminal program can be used to observe these commands or to send the reset “R” command to the Arduino. Feel free to contact the author via email at billv923@outlook.com for additional help.