Pinewood Derby Timer (PDT)

Software Configuration

and

PD Test/Tune/Track (PDT3) Utility

User Documentation

Last revised: 8 Mar 2016

http://www.miscejunk.org/pdt
Introduction

This document deals with the software side of constructing a Pinewood Derby Timer (PDT). The following subjects will be covered:

- setting up the Arduino environment
- configuring the timer's options
- verifying and uploading the timer sketch onto an Arduino Uno (or equivalent) board
- timer operation
- interfacing the timer with race management software on a PC
- using the free PD Test/Tune/Track (PDT3) software utility

Information on constructing the timer's hardware (circuit, bridge, etc.) can be found on the PDT web page:

http://www.miscjunk.org/pdt
Arduino Environment

The Arduino environment consists an integrated development environment (IDE) that allows programs to be edited, compiled and uploaded to the microcontroller (board). Refer to the Arduino web page which corresponds to your system type and follow the steps to download, install and test the IDE:

- **Mac OS:** [https://www.arduino.cc/en/Guide/MacOSX](https://www.arduino.cc/en/Guide/MacOSX)
- **Linux:** [https://www.arduino.cc/en/Guide/Linux](https://www.arduino.cc/en/Guide/Linux)

**STOP **** Do not continue until you have successfully uploaded the blink example to your board as described in the above instructions. If you could not get the blink example to work you will not be successful with the PDT sketch.

Adafruit Display Libraries

If your PDT build will include the optional Adafruit 7 segment displays you will need to install the libraries needed to drive the displays. Information about installing libraries can be found on the following web page:


Referring to the first section, “How to Install a Library using the Library Manager”, search for and install the following two libraries:

1. Adafruit GFX Library
2. Adafruit LED Backpack Library

PDT Sketch

If you have not done so already download the PDT sketch from the PDT website:


Create a folder (it can be anywhere on your computer) named `timer` and place the PDT sketch, named `timer.ino` into the folder.

**>>> NOTE >>>** The folder name MUST match the name of the PDT sketch. You can change the name but they must match. So if you name the sketch `fred.ino` then the name of the folder must be `fred`.

[http://www.miscjunk.org/pdt](http://www.miscjunk.org/pdt)
Timer Configuration

In the Arduino software open the PDT sketch `timer.ino` that you downloaded above. This section will describe the available configuration options for the timer.

The top of the PDT sketch contains the most common options that can be configured:

```c
/*-----------------------------------------*
- TIMER CONFIGURATION -
*-----------------------------------------*/
#define NUM_LANES    1                 // number of lanes
//@define LED_DISPLAY  1                 // Enable lane place/time displays
#define SHOW_PLACE   1                 // Show place mode
#define PLACE_DELAY  3                 // Delay (secs) when displaying time/place
#define MIN_BRIGHT   0                 // minimum display brightness (0-15)
#define MAX_BRIGHT   15                // maximum display brightness (0-15)
#define GATE_RESET   0                 // Enable closing start gate to reset timer
/----------------------------------------------------------*
- END -
*----------------------------------------------------------*/
```

Changing Number of Lanes

The default version of the PDT is configured for a one lane track. If you have a multiple lane track you will need to update the PDT sketch. The NUM_LANES variable should be changed from the default value of 1 to match the number of lanes on your track:

```c
#define NUM_LANES    1                 // number of lanes
```

Optional Adafruit Displays

The next five lines are used to configure the optional Adafruit displays. The default setup of the PDT is configured to not have any displays – the results are only sent to an attached PC.

If you choose to utilize the optional lane place/time displays use the following parameters to enable and configure them:

Enable Displays

The Adafruit libraries must be installed as described in the Arduino environment section above. To enable the displays the following line needs to be uncommented (remove the two slashes at the beginning of the line) so that it changes from this:

```c
//@define LED_DISPLAY  1                 // Enable lane place/time displays
```

To this:

```c
#define LED_DISPLAY  1                 // Enable lane place/time displays
```

Display Behavior

By default, when utilizing the display functionality, as cars finish the race their finishing place

http://www.miscjunk.org/pdt
is displayed. When all cars have finished the displays will then alternate between the finish order and elapsed time. To change the time delay between the two change the following parameter to desired number of seconds:

```c
#define PLACE_DELAY 3           // Delay (secs) when displaying time/place
```

If it is desired to only show elapsed time – with no finish order – the following line should be changed from the default,

```c
#define SHOW_PLACE 1            // Show place mode
```

to this

```c
#define SHOW_PLACE 0            // Show place mode
```

**Display Brightness**
The following two parameters limit the range of the dimmer knob which controls the brightness of the lane displays. Limiting the maximum brightness can control (usually to minimize) the total power draw of the lane displays.

```c
#define MIN_BRIGHT 0            // minimum display brightness (0-15)
#define MAX_BRIGHT 15            // maximum display brightness (0-15)
```

**Gate Reset**
The following parameter (when set to 1) enables the gate reset functionality, the default is disabled (set to 0). When enabled the timer – when in the FINISHED state – can be reset by closing the start gate. Caution should be taken if manually recording results – resetting the timer clears the displays.

```c
#define GATE_RESET 0            // Enable closing start gate to reset timer
```

**Status Light (LED) Troubleshooting**
During testing or operation of the timer you might notice that the green and blue status colors seem to be swapped. This can happen because the supplier of the LED keeps switching the source part and the pin orders on the components vary.

This can be fixed by changing the 3 lines in the 'pin assignments" section of the sketch from,

```c
byte STATUS_LED_R =  9;        // status LED (red)
byte STATUS_LED_B = 10;        // status LED (blue)
byte STATUS_LED_G = 11;        // status LED (green)
```

to the following (the only change is swapping the '10' and '11'),

```c
byte STATUS_LED_R =  9;        // status LED (red)
byte STATUS_LED_B = 11;        // status LED (blue)
byte STATUS_LED_G = 10;        // status LED (green)
```

>>> NOTE >>> Don't forget to save your changes!

http://www.miscjunk.org/pdt
Upload the PDT sketch to the Arduino board

Once the timer configuration is completed upload the PDT sketch to the board by pressing the 'Upload' button in the Arduino software. This is the same process used when you tested your environment by uploading the blink example earlier.

http://www.miscjunk.org/pdt
Manual Operation of the PDT

Manual operation of the PDT is simple, the current status of the timer is indicated by the color of the status LED.

READY (Blue LED)
The timer is ready for a race to start. If displays are used they will each show dashes ‘-----’ indicating the PDT is ready.

Hitting the reset button the PDT (with the start gate closed) puts the timer into the 'READY' state.

Starting the race by opening the start gate (either via software or manually) causes the timer to leave the 'READY' state and enter the 'RACING' state.

RACING (Green LED)
The race has started and the timer is watching each lane for cars to finish. If displays are used they will each go blank indicating the race is in progress.

If displays are used, as cars finish the displays will show either their current place (1\textsuperscript{st}, 2\textsuperscript{nd}, etc.) or their elapsed time (in seconds) depending on how the timer was configured.

Once all cars have finished the timer will leave the 'RACING' state and enter the 'FINISHED' state.

Hitting the reset button while in the 'RACING' state will cause the current race to end and will put the timer in the 'FINISHED' state.

FINISHED (Red LED)
The race is over. If displays are used they are either alternating between place and time or simply showing the time (based on the timer configuration).

Hitting the reset button (with the start gate closed) will cause the timer to leave the 'FINISHED' state and enter the 'READY' state. If the start gate is open when the reset button is hit the timer will remain in the 'FINISHED' state. If the Gate Reset option is enabled the timer will be reset automatically when the start gate is closed.

>>> NOTE >>> If the blue and green status colors seem to be swapped see the 'Status Light (LED) Troubleshooting' information in the earlier 'Timer Configuration' section.
Interfacing the PDT with software - GrandPrix Race Manager

The PDT can be interfaced with the popular GrandPrix Race Manager software. To configure GPRM for use with the PDT within GPRM select the “Options” menu, then select “Setup Options” and finally select “Hardware Options…”.

On the resulting “Hardware Setup” window on the “Timing tab”, select “Other Serial Timer” in the “Timing System” group then hit the “Custom Setup” button. On the resulting “Custom Serial Timers Options” window enter the following configuration and then hit the “Save” button.

<table>
<thead>
<tr>
<th>COM Port Settings</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud:</td>
<td>9600</td>
</tr>
<tr>
<td>Data Bits:</td>
<td>8</td>
</tr>
<tr>
<td>Parity:</td>
<td>None</td>
</tr>
<tr>
<td>Stop Bits:</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Start Gate Status</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Command:</td>
<td>G</td>
</tr>
<tr>
<td>Open Response:</td>
<td>O</td>
</tr>
<tr>
<td>Timer Start Message:</td>
<td>B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Timer Reset</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset Command:</td>
<td>R</td>
</tr>
<tr>
<td>Ready Response:</td>
<td>K</td>
</tr>
<tr>
<td>Response Delay:</td>
<td>0.25 secs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lane Masking</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask Command:</td>
<td>M</td>
</tr>
<tr>
<td>Remove all Masks:</td>
<td>U</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other Software Commands</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Force Data Send:</td>
<td>F</td>
</tr>
<tr>
<td>Trigger Solenoid:</td>
<td>S</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Miscellaneous</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Read Delay:</td>
<td>0.25 secs</td>
</tr>
<tr>
<td>Precision:</td>
<td>0.0001 secs</td>
</tr>
<tr>
<td>Max Time:</td>
<td>9.9999 secs</td>
</tr>
<tr>
<td>Lane Labels:</td>
<td>Numbers</td>
</tr>
</tbody>
</table>

Refer to the GPRM help for general usage of the application.

http://www.miscjunk.org/pdt
Using the PD Test/Tune/Track (PDT3) Utility

The PDT3 utility is a free software program that fully interfaces with the PDT and allows you to record, analyze and archive car run times during the testing/tuning process of building a Pinewood Derby car.

Configuration

After installation a shortcut for PDT3 will be placed on your desktop similar to the one shown here. If needed the PDT3 utility can be uninstalled via the Control Panel → Programs and Features window.

The first step to perform after installation is to setup the connection between the PDT and the PDT3 utility. Selecting “Options...” from the “Edit” menu will display the following dialog:

>>> NOTE >>> You might experience a long delay when this window is opened or when you attempt to select a COM port. This is caused by a bug in the programming language that was used to create this utility. The window will eventually respond and return to normal.

In the “Timer” group select the COM port for the PDT from the list of available ports and hit the “Connect” button. If successful you will see the COM port displayed in the status bar of the main PDT3 screen.

IMPORTANT: The timer needs to be in the 'READY' or 'FINISHED' state when starting PDT3 or manually connecting. If the timer is in the 'RACING' state PDT3 will not be able to connect.

Note: The number of lanes does not need to be configured, the software obtains that information from the timer.

http://www.miscjunk.org/pdt
The last step to perform before first use is to create a new database file by selecting “New” from the “File” menu. Typically each car would have its own database file. The current database file is listed in the status bar of the main screen.

**Current Session Tab**

The main screen of the PDT3 utility, the Current Session tab, is shown below. This screen displays and records what is considered the current working session. As runs are performed the run times are added to the run list at the right. The overall average and standard deviation are automatically calculated as runs are added. In addition the delta time difference between each run time and the overall average is calculated for each run.

The Lane selection list specifies which time is captured on a multi-lane track. Whether this value can be changed after the first run is performed is controlled by a flag in the “Option...” dialog.

The “Notes” field is a free-form field available for a description of the session.

Erroneous run times can be deleted by selecting the run time in question and hitting the

[http://www.miscjunk.org/pdt](http://www.miscjunk.org/pdt)
delete button to the lower right (X). The entire list of run times can be copied to the clipboard (in csv format) by hitting the copy button to the upper right ( ).

The date/time displayed above the run time list reflects when the session was created. The “New Session” button will archive the current session data and start a new one.

**Track / Timer Control**

The Track/Timer control (shown below) on the “Current Session’ tab is used to control the PDT. It will only be active if there is a successful connection to the PDT.

![Timer Control](http://www.miscjunk.org/pdt)

The “Reset” button sends a reset message to the PDT. The status field indicates the current status of the track/timer. Description of timer/track status:

- **Not Ready** - the track/timer is not ready (need to reset)
- **Gate Open** - the start gate is open (cannot reset until closed)
- **Ready** - the timer/track is ready (to start)
- **Racing** - the race is in progress
- **Finished** - the race is completed

The “Start” button will send a start message to the PDT. If the PDT is equipped with the optional start gate solenoid circuit this message will open the start gate. If the PDT does not have that option simply initiate the run by manually opening the track’s start gate – the software will automatically determine that the run has been started.

The “Force End” button will send a force end message to the PDT which will immediately send all lane times to the PDT3 software. This feature is typically used when for some reason (a derailment, for example) the car does not finish the run.

**Lane Times Window**

The “Lane Times” window, shown here, can be opened by selecting “Lane Times” from the “View” menu. This window display the run times for each lane at the completion of the run.
Lane Masks Window

The “Lane Masks” window, shown here, can be opened by selecting “Lane Masks” from the “View” menu. On a multi-lane track this window allows you to mask unused lane(s). The “Unmask All” button resets (clears) all lane masks.

Masking a lane tells the PDT to not wait for a car to finish on an unused lane (since there is no car on that lane or lanes).

History tab

The “History” tab, shown below, allows you to search for, browse and display all historical sessions. Data cannot be altered but sessions or runs can be deleted.