Lukasz Bugaj

Bugaj Lab

November 18, 2018

***User Manual***

*optoPlate-96\_1colorPulse.ino*

*optoPlate-96\_2colorPulse.ino*

*optoPlate-96\_3colorPulse.ino*

These scripts control the optoPlate-96, allowing complete control of illumination parameters (timing and intensity) for up to 3 colors for each of 96 positions.

NOTE: Although this manual refers to illumination wavelengths for the 3-color optoPlate configuration (for blue, red, and far-red illumination), the manual applies to all configurations. 1- and 2-color configurations will simply have fewer light intensity variables to specify.

The optoPlate-96\_XcolorPulse.ino scripts provide a simple yet powerful framework for dynamic optogenetic perturbations, as described below. However, the optoPlate-96 may be programmed for arbitrarily complex illuminations through customization of the code.

To run this code, one custom library (TLC5947\_optoPlate) must be installed. A zipped file of the library can be found in the [GitHub repository](https://github.com/BugajLab/optoPlate-96/tree/master/optoPlate_library) (filename: TLC5947\_optoPlate.zip). For help installing custom libraries, visit: <https://www.arduino.cc/en/Guide/Libraries>.

*Note: this library is a modification of the Adafruit TLC5947 library, found here: https://learn.adafruit.com/tlc5947-tlc59711-pwm-led-driver-breakout/downloads-and-links*

***Phases of experiment***

The general illumination program consists of 4 programmable phases that allow pulsing cells between 2 experimental illumination conditions (Figure 1A):

0: *phaseDelay:* the amount of time to delay the start of illumination. This is primarily useful in timecourse experiments when staggered starts are required.

1: *phase1:* the first illumination condition (can be thought of as the ON phase)

2: *phase2:* the second illumination condition (or, the OFF phase)

There is also an OFF phase *(phaseOFF)*, in which all LEDs are turned off.

For each phase 0-2, the user can define the duration and intensity of each illumination color. The duration of each phase is specified for each individual LED (*phaseTimes*). Intensities within each phase can be specified globally for all LEDs or locally for each individual LED.

During an experiment, the LEDs will display the defined illumination conditions sequentially (*phaseDelay -> phase1 -> phase2*) for the durations specified. Once the end of *phase2* has been reached, illumination cycles back to *phase1*, and the program oscillates between *phase1* and *phase2* indefinitely following the defined phase timings. If only illumination in *phase1* is desired (only ON illumination), then the duration of *phase1* should be set arbitrarily large (longer than the desired experiment). Similarly, if the experiment requires that illumination progresses through *phase1* and *phase2* exactly once, then the duration of phase2 should be defined arbitrarily large.

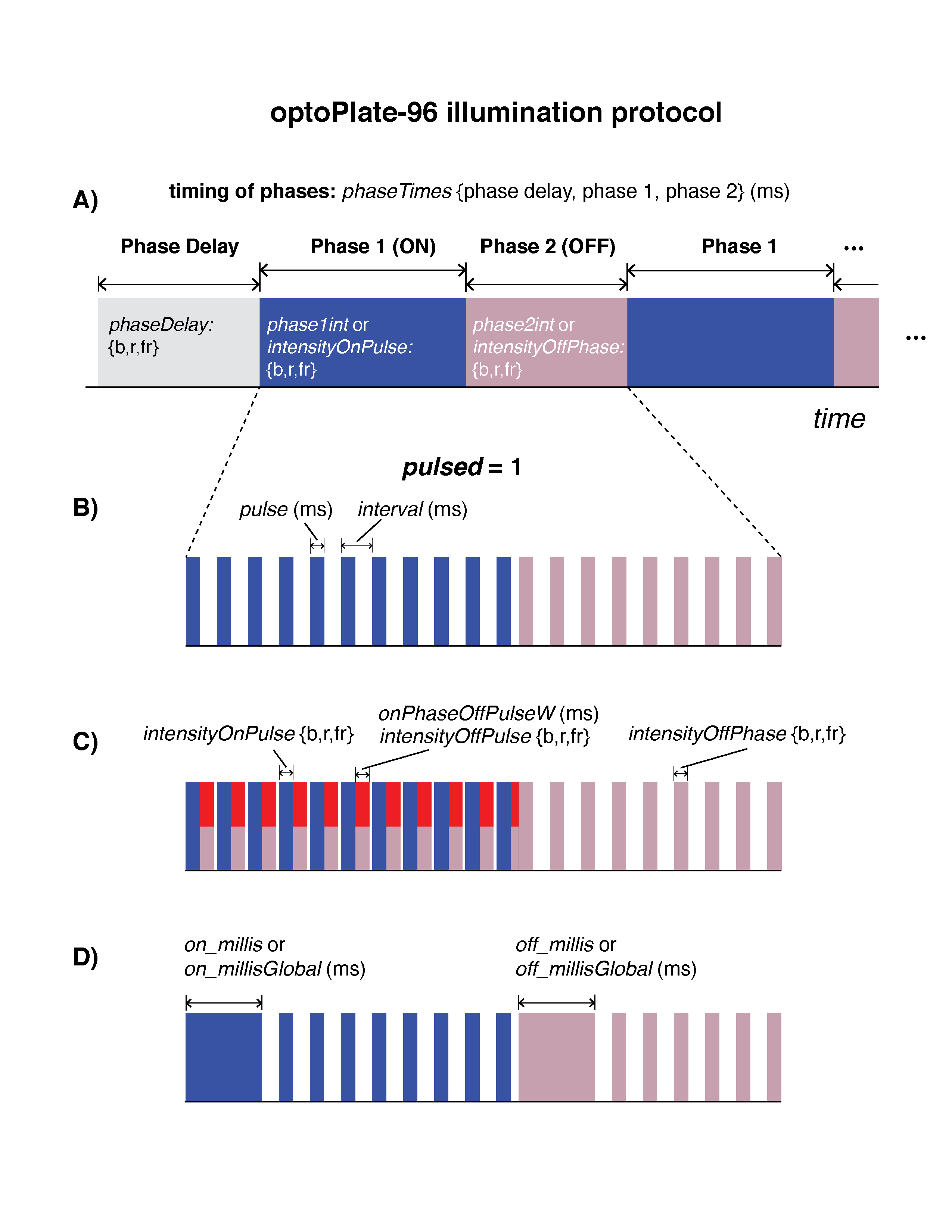
***Pulsing***

Within each phase, illumination can be either pulsed or constant (Figure 1B). Pulsed illumination is recommended to minimize both cellular phototoxicity and heat generation from the optoPlate illuminator (heat scales with the amount of power used – or, the total light output). Pulsing parameters—pulse width (*pulse)* and pulse period (*interval*)—can be defined for each individual LED. Note that these parameters will dictate pulsing across all phases of illumination (*phaseDelay, phase1, phase2*) (Figure 1B).

Within *phase1*, there is also the option to specify illumination during the OFF pulse of (the duration between pulsing of *phase1*-defined illumination settings, Figure 1C). This is mostly useful for three-color optogenetic control of red- and blue-sensitive optogenetic tools (eg, PhyB and iLid), where blue light (ON pulse) must be followed by far-red light (OFF pulse) to prevent blue-light cross-activation of PhyB. The intensity and duration of this OFF pulse are customizable, though the OFF pulse duration should not exceed the amount of time between ON pulses (eg, if the interval is 5s and the ON pulse is 2s, OFF pulse illumination should be < 3s).

***Transitions between phases***

When the program transitions to *phase1* or *phase2* and pulsing is used, the ON->OFF or OFF->ON transitions of optogenetic tools can be slower than desired. To speed the activation or inactivation of optogenetic probes, the user can define a length of time (*on\_millis, off\_millis)* during which the new illumination setting will constant (not pulsed)(Figure 1D). This results in more step-like (in)activation of the optogenetic probe, and the new state can be further maintained through pulsed illumination. This timing parameter can be defined individually for each position or globally for all wells (*on\_millisGlobal, off\_millisGlobal*).

**Figure 1** Visualization of optoPlate-96 illumination protocols and variable definitions. Color of bars indicates colors that would be used in a typical 3-color optogenetic experiment. In the examples provided, *phaseDelay* could have no illumination, or could illuminate with inactivating far-red light. *phase1* would illuminate activating blue or red light, and compensatory far-red light to prevent blue light cross-activation of PhyB. *phase2* would illuminate with inactivating far-red light.

A) Schematic of experimental phases. B) If *pulsed = 1, pulse* and *interval* define the pulsing parameters. C) Intensity variables are depicted. For 3-color experiments, *onPhaseOffPulseW* and *onPhaseOffPulse* allow definition of intensity and timing illumination between activating pulses. D) *on\_millis* and *off\_millis* define the timing of constant illumination at the start of a new illumination phase. This enables more step-like (vs gradual) transitions between phases.

The script is divided into 3 sections:

**SECTION 1:** This section defines all the variables for each experiment. Variables that need not be changed are labeled \*DO NOT CHANGE\*. Below are notes on important variables to define:

*Time variables:* time variables are defined globally, and pointers are used throughout the script to reference these variables. For example, *&m30* and *&s30* are used to point to the variables that specify 30 minutes and 30 seconds, respectively. Additional time values can be added as new variables in the first section. When adding new variables, make sure to copy the syntax of the other variables. Also, when adding a new time variable, add a corresponding variable in Section 2 in the “test mode” loop. This will ensure that this time variable is properly scaled in test mode.

*LEDmode:* this defines which of the 96 positions on the LED illuminator are operational. Values are laid out in 96-well format and correspond to LED layout.

*Note: arrays of values are most easily defined in a spreadsheet (eg Excel) and then copied and pasted into the Arduino environment*

*phaseTimes:* defines the duration of each phase for each of the 96 LED positions. This variable in particular is most easily defined in a spreadsheet and copied over.

*useVarOnPulseIntensity*

*useVarOffPulseIntensity*

*useVarOffPhaseIntensity:* determine whether LED intensities will be globally defined (0) or variably specified (1) for each individual LED.

*intensityOnPulse*

*intensityOffPulse*

*intensityOffPhase:* define intensities for each color for individual LEDs. First define 96 blue values, then 96 red values, then 96 far-red values.

*pulsed:* define whether LED illumination will be pulsed (1) or constant (0)

*interval, pulse, onPhaseOffPulseW:* define pulse periods and pulse widths. Here (and only here), use integer values of milliseconds (do not use pointers to the time variables defined earlier)

*varOn, varOff:* if constant illumination will be used after transitioning between phases, define whether this will be globally specified (0) or individually for each LED (1).

*test:* define whether to run the protocol in test mode (*test = 1*). Test mode allows you to rapidly verify that you have correctly programmed your experiment by speeding up the program by a defined factor. It is *strongly* recommended to run every protocol in test mode before each experiment to verify correct programming. In test mode, LED illumination is constant (*pulsed = 0*).

*factor:* define by what factor to speed up the experiment in test mode.

*fanSpeed:* define the speed of the fan on the heatsink. The faster the fan, the less sample heating occurs during the experiment. Fan speed can be set from ~100-255. Lower speeds (<100) may result in no fan spinning. Generally, max speed (255) is recommended.

**SECTION 2:** This is the setup function. This sets the initial settings for the optoPlate run.

**SECTION 3:** This is the loop function. This code executes the optoPlate experiment as defined by the variables in Section 1.