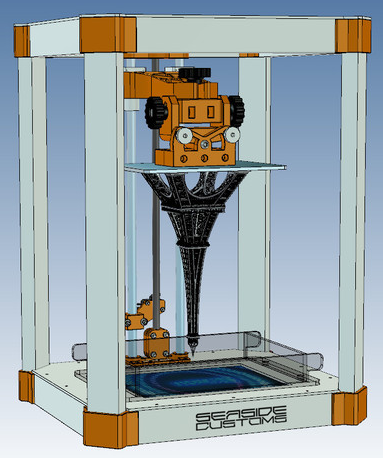
**3DSim – Virtual 3D Printer Datasheet**

**Hardware specifications**

Build-plate dimensions: **200mm x 200m**

Build-plate origin (0,0): **center of plate**

Maximum print height: **100mm**

Z coordinate at build surface: **0.00 mm**

Max. distance between resin tray and build plate at start of print: **0.55mm**

Stepper steps per mm: **400**

Stepper maximum velocity: **40mm/s**

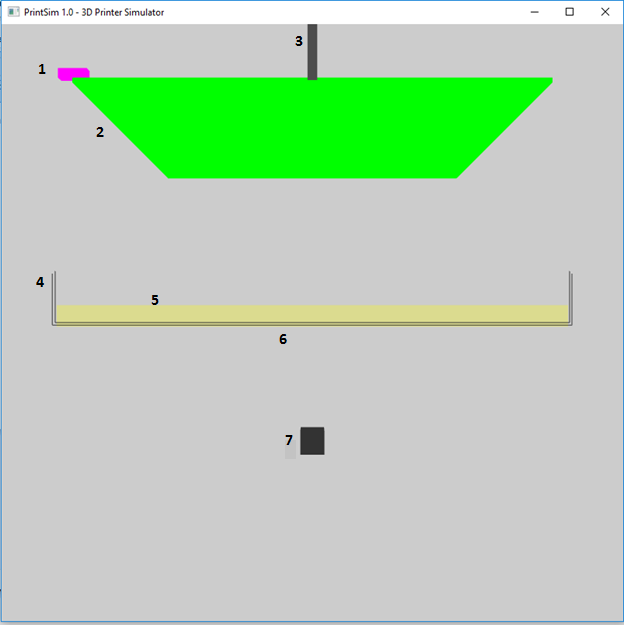
Stepper maximum acceleration: **4 mm/s**

Galvanometer voltage: **+/- 2.5 volts**

Host communication: **Serial (see APIs below)**

Serial buffer size: **64 bytes**

continued on reverse side



1. **Limit switch** – indicates when build plate is at its maximum extent
2. **Build plate** – the surface upon which the model is built
3. **Z-rail axis** – this rail is connected to a stepper motor that raises and lowers the build plate
4. **Resin tray** – the clear tray used to contain the photosensitive resin
5. **Resin** – the resin that is cured by the laser
6. **Build surface** – this interface between the clear resin tray and the resin is where all building occurs
7. **Galvos/laser** – the galvanometer/laser assembly that is used to direct the laser on a 2D plane to cure resin on the build plate.

**Low-level API functions – Accessible ONLY by HOST**

public void Start()

Initialize virtual printer – This API must be called before accessing other APIs

public void Stop()

Shutdown virtual printer

public void WaitForInit()

Wait until the virtual printer is fully initialized and ready for commands

public int WriteSerialToFirmware(byte[] Buffer, int bytesToWrite)

Sends a buffer of bytes to the firmware over the serial communications link. Returns number of bytes successfully written. NOTE: The serial communication link between the host and the firmware is sketchy. Sometimes data will get corrupted. You will need an error checking command protocol.

public int ReadSerialFromFirmware(byte[] Buffer, int bytesRequested)

Reads the desired number of bytes from the firmware. Returns the number of bytes actually read. If the number of requested bytes is not available, returns 0.

**Low-level API functions – Accessible ONLY by FIRMWARE**

public int WriteSerialToHost(byte[] Buffer, int bytesToWrite)

Sends a buffer of bytes to the host over the serial communications link. Returns number of bytes successfully written.

public int ReadSerialFromHost(byte[] Buffer, int bytesRequested)

Reads the requested number of bytes from the host. Returns the number of bytes actually read. If the number of bytes requested is not available, returns 0.

public bool LimitSwitchPressed()

Returns true when the limit switch at the maximum build-plate height is pressed or false if it is not pressed.

public void WaitMicroseconds(long uS)

Waits for the desired number of microseconds.

public void ResetStepper()

Resets the printer’s Z-rail stepper if an error occurs.

public bool StepStepper(StepperDir dir)

Steps the Z-rail stepper one step in the desired direction. Possible directions are StepperDir.STEP\_UP and StepperDir.STEP\_DOWN.

public void MoveGalvos(float xVoltage, float yVoltage)

Moves the X and Y galvanometers to the positions indicated by the voltages values. Voltages of 0,0 indicate the center of the build plate. An X voltage of -2.5 indicates the left edge of the build plate, and 2.5 indicates the right. The same goes for the Y axis.

public void RemoveModelFromPrinter()

Removes the model from the build plate after a build is completed. This must be done in order to build another model on the printer.

public void SetLaser(bool fOn)

Turns the curing laser on and off.