PROGRESS REVIEW 1

Astha Prasad

Team F / ADD_IN

Teammates: Daniel Berman, Nikhil Baheti, Ihsane Debbache

ILR #2

October 22nd, 2015

Individual Progress

For this week, my work was mostly focused on two parts:

- Modifying slicer code to enable insertion layer selection
- Updating the website

Slic3r

This week's software goal involved adding the necessary G-Code commands to the code generated by Slic3r. This modification would enable the user to select a height at which the printer will pause, thus making way for safe insertion of the COTS part. In order to achieve this goal, the first step was to download Slic3r from source. Dan figured out a way to invoke Slicer using MATLAB which made issuing Slic3r commands very easy and quick.

- Approach 1

Slic3r has a command called 'cut' using which, it is possible to cut an STL file into 2 parts at a certain 'z'. The 'z' height is provided by the user along with the STL file. Two separate G-code files can be generated using this command. Dan wrote a MATLAB script that read in an STL file and invoked the 'cut' feature, prompting the user to enter the height. Using this script, we were able to generate two g-code files, namely, 'upper.gcode' and 'lower.gcode'. At first, we tried this approach, hoping that we would be able to stitch together the G-Codes after adding the necessary pause commands. However, after closer inspection, I found that this approach was rather complicated since the G-Code file of the upper part was generated assuming it would be printed from the base of the bed (z = 0), i.e., as a new print. This approach added several lines of unnecessary code at the end of lower.gcode and the beginning of upper.gcode which included:

<u>End of Lower.gcode</u>

```
M104 S0;      ( turn off temperature)
G28 X0;       ( home X axis )
M84;          ( disable motors)
```

<u>Beginning of Upper.gcode</u>

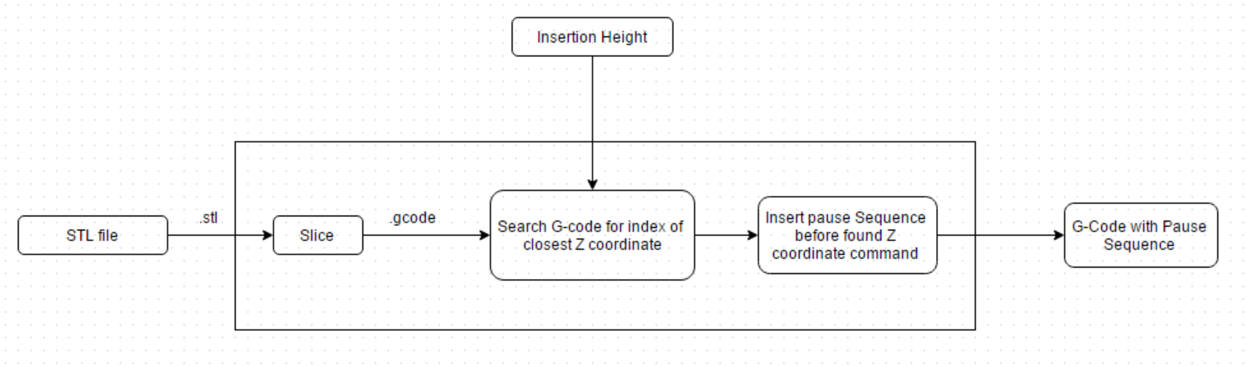
```
M107;             (Turn Fans off)
M104 S200 ;       ( Set temperature)
G28 ;             ( home all axes)
G1 Z5 F5000 ;     ( lift nozzle )
M109 S200 ;       ( wait for temperature to be reached)
```

G21 ;	(set units to millimeters)
G90 ;	(use absolute coordinates)
M82 ;	(use absolute distances for extrusion)

In addition to these lines of code, all the Z coordinates of upper.gcode had to be incremented by the layer height entered by the user. Due to these challenges, we decided to work with only one Gcode of the entire object.

- Approach 2

This approach follows the following path:


Insertion Height
STL file
.stl
Slice
.gcode
Search G-code for index of closest Z coordinate
Insert pause Sequence before found Z coordinate command
G-Code with Pause Sequence

As shown, we find the location of the command containing the Z coordinate closest to the height entered by the user. This is done by first searching for all commands containing a Z coordinate and storing their start and end indices. All such commands are of the form ‘G1 Z#.#’. This search is done using the following function:

```
[startIndex,endIndex] = regexp(Gcode,'G1\sZ\d*\.\d*');
```

Once we have the start and end indices, we can extract the #.# from each Z command and store them in a separate array. We then compare the entered height with each of these heights in the array and find the index of the first Z coordinate that is larger. Using the corresponding start and end location in the G-code, we insert a pause sequence in between which is as follows:

```
pauseSequence = 'G01 X0 Y200 Z170
                 M0';
```

The first line moves the nozzle and bed to a safe insertion configuration by pushing the hot nozzle to the back end of the printer and rolling the bed all the way forward and out. The printing is paused using the command ‘M0’. This is an unconditional stop. When it is encountered, the machine finishes any moves left in its buffer, then shuts down. All motors and heaters are turned off. Once the COTS part has been inserted, the printer can be started again by pressing the reset button on the master microcontroller.

Updating the Website

Since I am in charge of Website management and the website check was scheduled for 23rd October, a large part of my week was spent updating the various sections of the website that were relevant at this point of time.

Challenges

Using the printer this week was challenging since the hardware team was experimenting with the first prototype of the coupler that connected the extruder and nozzle at an angle of 45 degrees. This meant that the height of the extruder-nozzle set up was much larger than what it usually is. As result, the G-code that directed the nozzle to its 'home' position could not be used since the bed would hit and possibly break the extruder-nozzle setup. We worked around this by deleting the 'home' commands from the g-code every time the code was tested on the printer.

Another challenge was not being able to get the Slic3r GUI working. Since the Slic3r settings for our printer(such as filament diameter, extruder diameter, layer height, etc.) are different from the default Slic3r settings, updating these settings using command prompt/MATLAB was a tedious task.

Teamwork

The team split up the tasks in the following manner:

Daniel Berman

Dan and I worked on the Slic3r subsystem together. Dan was responsible for finding a way to invoke Slic3r using MATLAB.

Astha Prasad

I worked with Dan on the MATLAB code and was responsible for figuring out a way to pause the print appropriately.

Nikhil Baheti & Ihsane Debbache

Nikhil and Ihsane together worked on the printing the Extruder-Nozzle Coupler. They used to MakerBot 3D printer in the lab to print the piece. Ihsane was in charge of the calculations required to select the most suitable rotary stage while. They had also printed a new extruder holder which was slightly incompatible with the nozzle. I buffed the holder so that both, the original nozzle and the coupler would fit into it.

Future Goals

Our goals for the next week are as follows:

1. Troubleshoot slic3r installation to get Wx module and the GUI working

2. Update MATLAB script to be more robust and allow multiple insertion layers selection

3. Load slic3r configuration file through MATLAB script to set advanced slic3r settings.

4. Work on the CAD files of the stepper motor, filament drive and extruder design. This is to integrate CAD part 3 with the project. The stepper motor CAD files will be helpful in designing the rotary joint.

5. Modify the extruder hot-end to make a bent section at the end of the hot-end so that the nozzle can be screwed on to it.

Among these, I will work on making the Slic3r MATLAB program more robust and also try to install the Slic3r GUI by troubleshooting the common error Dan and I are facing.

Additionally, I will work on the PCB schematic that is due this week.

MATLAB Code

```matlab
dansComputer = false;
if dansComputer
    cd '~/Slic3r'
    [file,path] = uigetfile('~/Desktop/*.stl','Select .stl File');
else
    cd 'C:/CMU/Courses/Project/Slic3r';
    [file,path] = uigetfile('../Desktop/*.stl','Select .stl File');
end

file = strrep(file,'.STL','.stl');

%TODO: Show part

usrInput = inputdlg('Enter Height');

%TODO: Show insertion plane

insertionZ = str2double(usrInput);

if dansComputer
    system(['./slic3r.pl ',[path,file]]);
else
    system(['slic3r.pl ',[path,file]]);
end

%Read in gcode files
Gcode = fileread(strrep([path,file],'.stl','.gcode'));
[startIndex,endIndex] = regexp(Gcode,'G1\sZ\d*\.\d*');

for i = 1:length(startIndex)
    Zlayers(i) = str2num(Gcode(startIndex(i)+4:endIndex(i)));
end

ind = find(Zlayers>insertionZ,1);
insertLoc = startIndex(ind);

pauseSequence = 'G01 X0 Y200 Z170\nM0\n';

Gcode = [Gcode(1:insertLoc-1),pauseSequence,Gcode(insertLoc:end)];

f = fopen([path,'FinalGcode.txt'],'w');
fprintf(f, Gcode);
fclose(f);
```