Additive Manufacturing – Module 4

Spring 2015

Wenchao Zhou

zhouw@uark.edu

(479) 575-7250

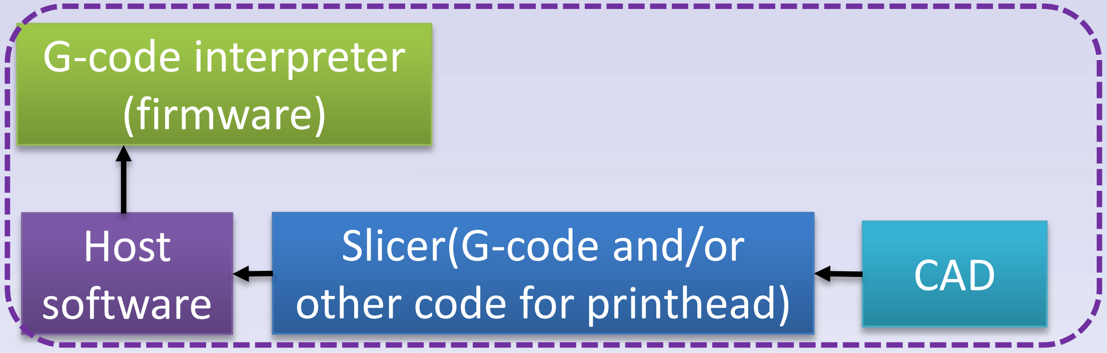
The Department of Mechanical Engineering
University of Arkansas, Fayetteville

UNIVERSITY OF ARKANSAS

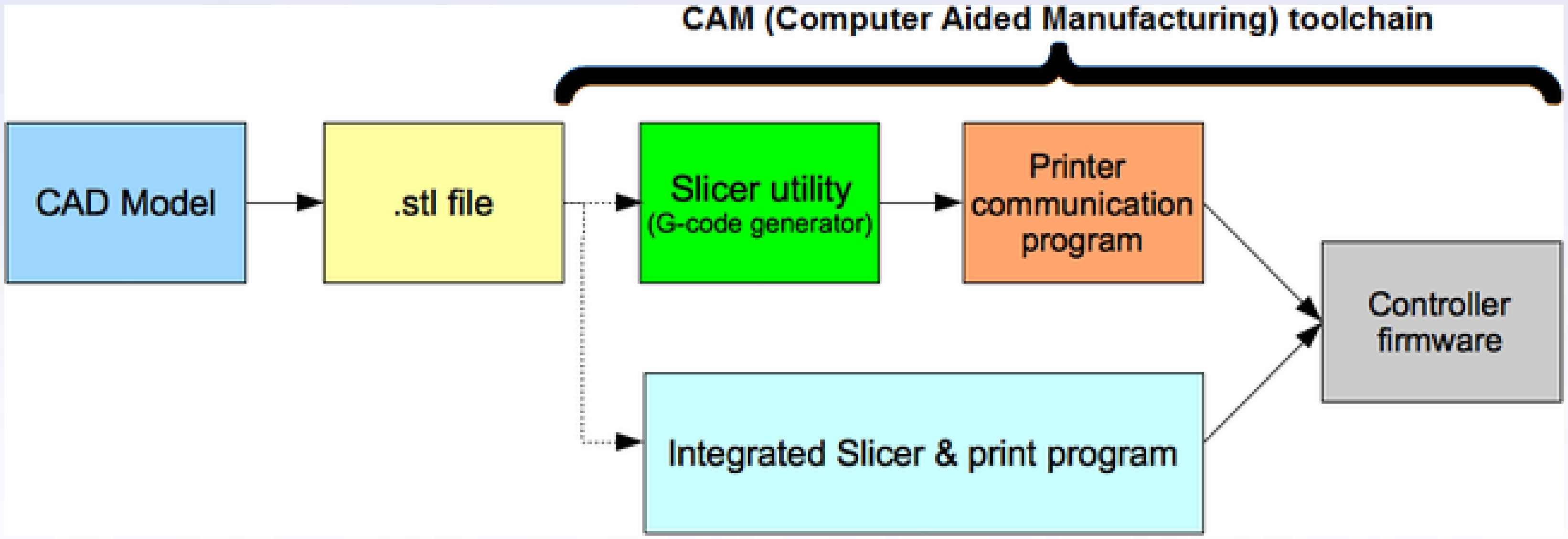
Software

AM³ Lab
Advanced Manufacturing | Modeling | Materials

Tool Chain


G-code interpreter (firmware)
Host software
Slicer(G-code and/or other code for printhead)
CAD

Software system


CAM (Computer Aided Manufacturing) toolchain
CAD Model
.stl file
Slicer utility (G-code generator)
Printer communication program
Controller firmware
Integrated Slicer & print program

UNIVERSITY OF ARKANSAS

Software

AM³ Lab
Advanced Manufacturing | Modeling | Materials

Tool Chain

Introduction

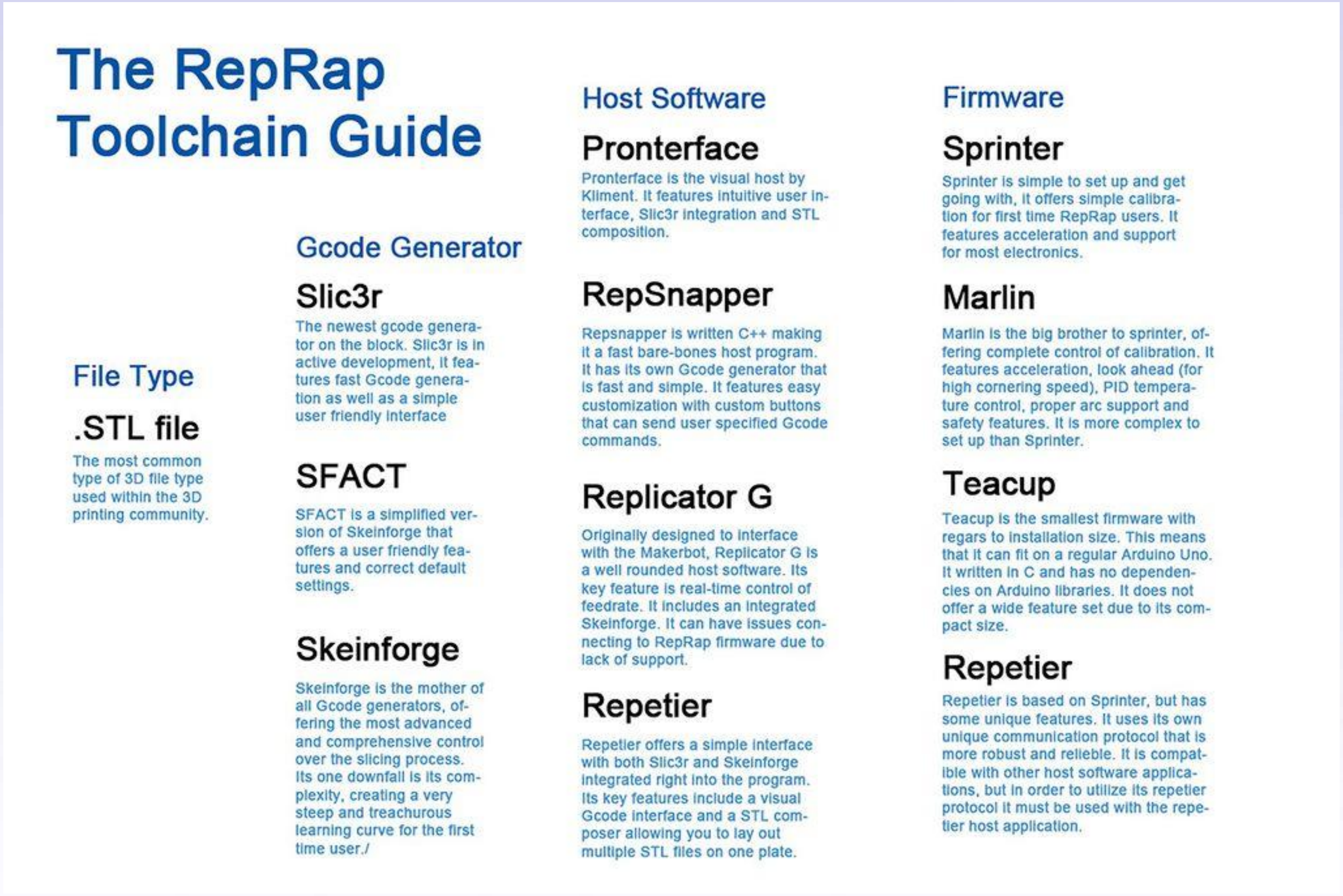
What

Why

How

Objectives

Syllabus



The RepRap Toolchain Guide

File Type

.STL file

The most common type of 3D file type used within the 3D printing community.

Gcode Generator

Slic3r

The newest gcode generator on the block. Slic3r is in active development, it features fast Gcode generation as well as a simple user friendly interface

SFACT

SFACT is a simplified version of Skeinforge that offers a user friendly features and correct default settings.

Skeinforge

Skeinforge is the mother of all Gcode generators, offering the most advanced and comprehensive control over the slicing process. Its one downfall is its complexity, creating a very steep and treachurous learning curve for the first time user./

Host Software

Pronterface

Pronterface is the visual host by Kliment. It features intuitive user interface, Slic3r integration and STL composition.

RepSnapper

Repsnapper is written C++ making it a fast bare-bones host program. It has its own Gcode generator that is fast and simple. It features easy customization with custom buttons that can send user specified Gcode commands.

Replicator G

Originally designed to interface with the Makerbot, Replicator G is a well rounded host software. Its key feature is real-time control of feedrate. It includes an integrated Skeinforge. It can have issues connecting to RepRap firmware due to lack of support.

Repetier

Repetier offers a simple interface with both Slic3r and Skeinforge integrated right into the program. Its key features include a visual Gcode interface and a STL composer allowing you to lay out multiple STL files on one plate.

Firmware

Sprinter

Sprinter is simple to set up and get going with, it offers simple calibration for first time RepRap users. It features acceleration and support for most electronics.

Marlin

Marlin is the big brother to sprinter, offering complete control of calibration. It features acceleration, look ahead (for high cornering speed), PID temperature control, proper arc support and safety features. It is more complex to set up than Sprinter.

Teacup

Teacup is the smallest firmware with regars to installation size. This means that it can fit on a regular Arduino Uno. It written in C and has no dependencies on Arduino libraries. It does not offer a wide feature set due to its compact size.

Repetier

Repetier is based on Sprinter, but has some unique features. It uses its own unique communication protocol that is more robust and relieble. It is compatible with other host software applications, but in order to utilize its repetier protocol it must be used with the repetier host application.

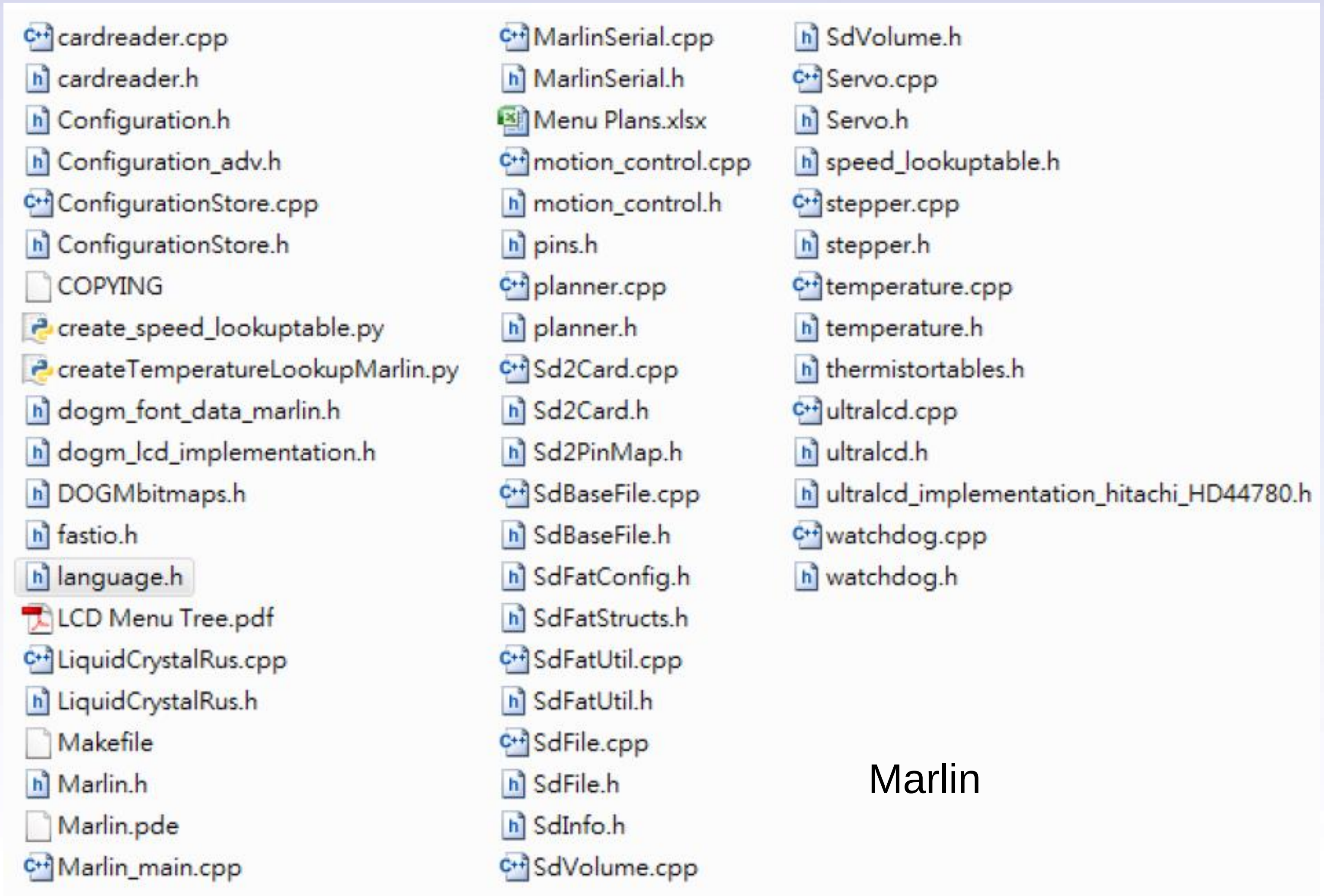
UNIVERSITY OF ARKANSAS

Software

AM³ Lab
Advanced Manufacturing | Modeling | Materials

Firmware

https://github.com/kliment/Sprinter
https://github.com/MarlinFirmware/Marlin
https://github.com/repetier/Repetier-Firmware



cardreader.cpp	MarlinSerial.cpp	SdVolume.h
cardreader.h	MarlinSerial.h	Servo.cpp
Configuration.h	Menu Plans.xlsx	Servo.h
Configuration_adv.h	motion_control.cpp	speed_lookuptable.h
ConfigurationStore.cpp	motion_control.h	stepper.cpp
ConfigurationStore.h	pins.h	stepper.h
COPYING	planner.cpp	temperature.cpp
create_speed_lookuptable.py	planner.h	temperature.h
createTemperatureLookupMarlin.py	Sd2Card.cpp	thermistortables.h
dogm_font_data_marlin.h	Sd2Card.h	ultralcd.cpp
dogm_lcd_implementation.h	Sd2PinMap.h	ultralcd.h
DOGMbitmaps.h	SdBaseFile.cpp	ultralcd_implementation_hitachi_HD44780.h
fastio.h	SdBaseFile.h	watchdog.cpp
language.h	SdFatConfig.h	watchdog.h
LCD Menu Tree.pdf	SdFatStructs.h	
LiquidCrystalRus.cpp	SdFatUtil.cpp	
LiquidCrystalRus.h	SdFatUtil.h	
Makefile	SdFile.cpp	
Marlin.h	SdFile.h	
Marlin.pde	SdInfo.h	
Marlin_main.cpp	SdVolume.cpp	

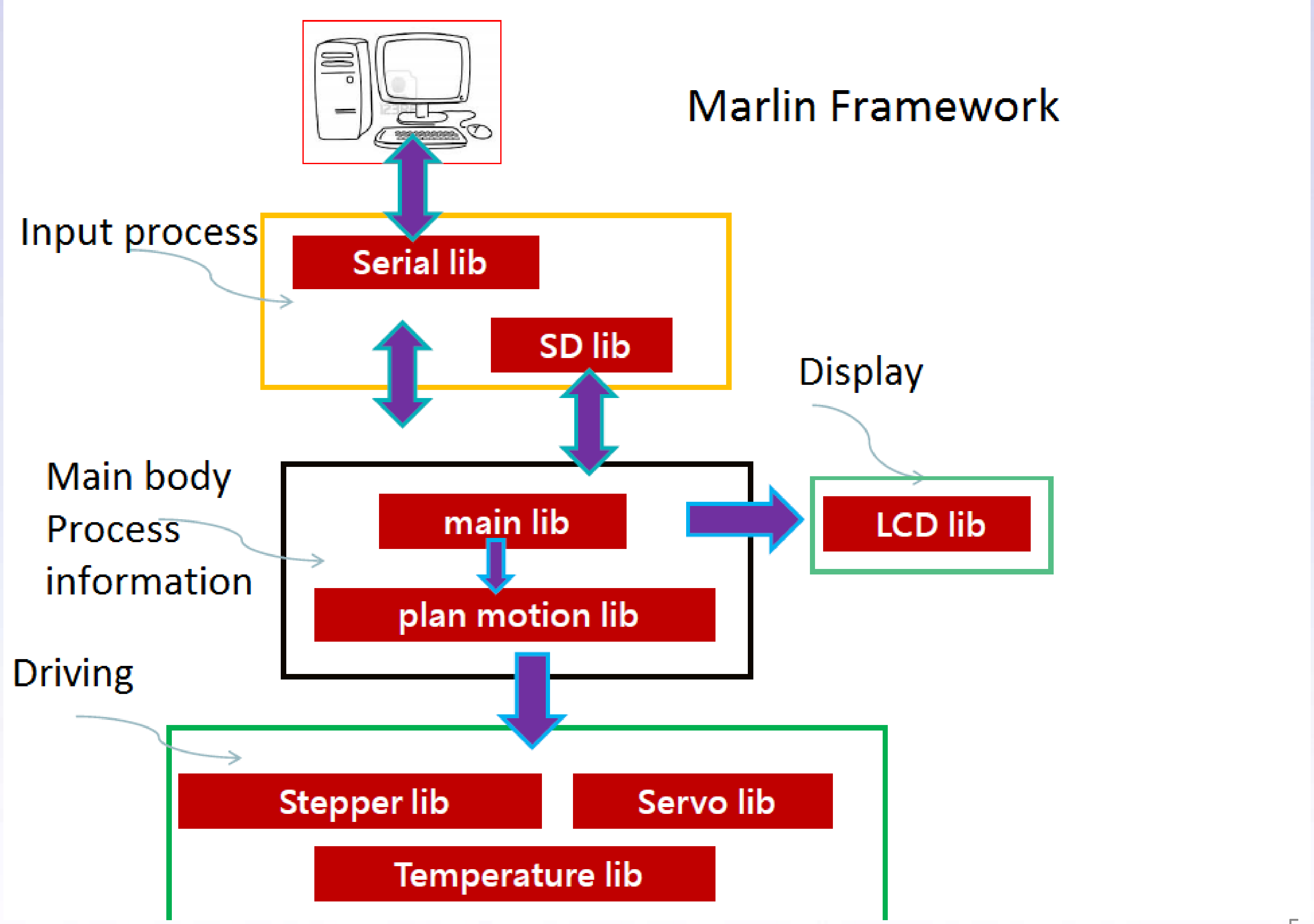
Marlin

Software

Advanced Manufacturing | Modeling | Materials

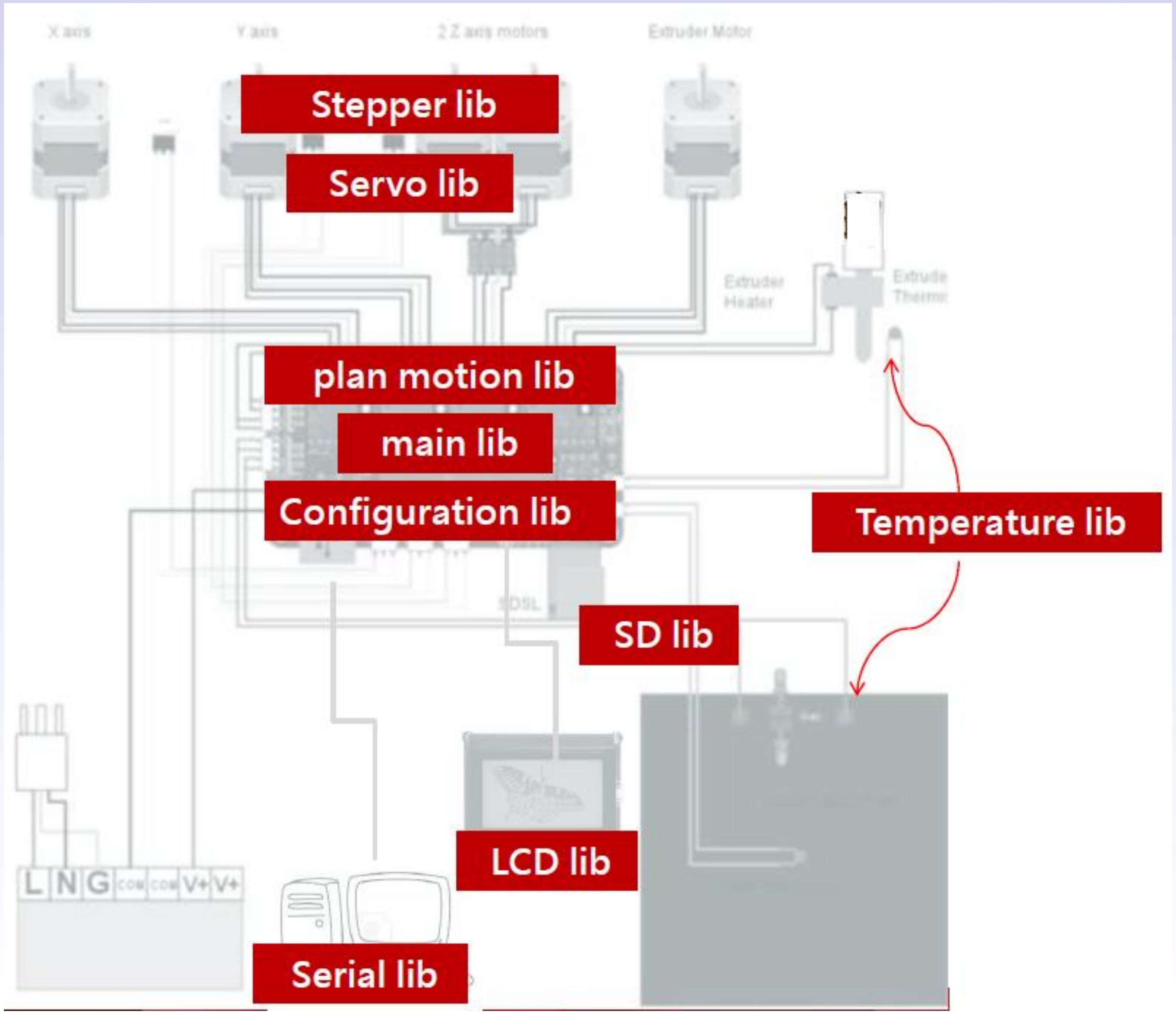
Firmware

Like an OS of Arduino



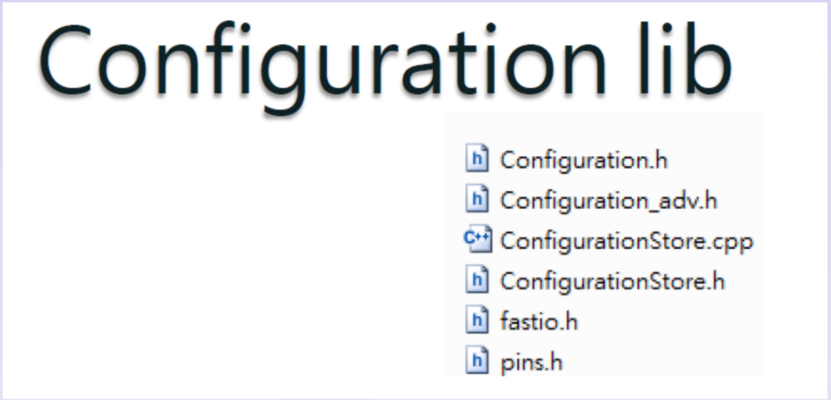
Software

Firmware



Software

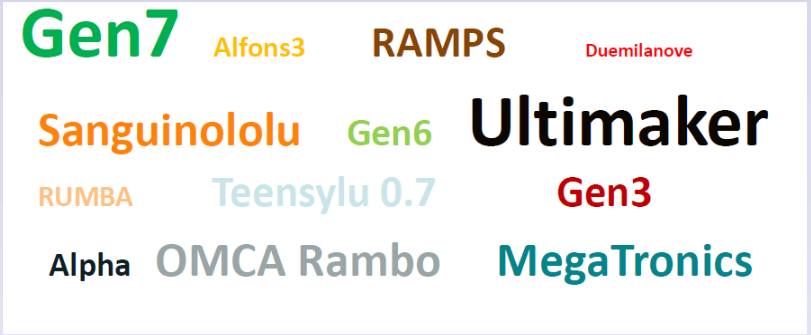
Firmware

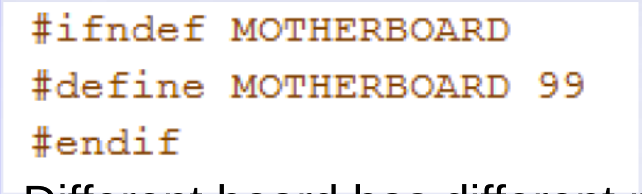


Configuration lib

- Configuration.h
- Configuration_adv.h
- ConfigurationStore.cpp
- ConfigurationStore.h
- fastio.h
- pins.h

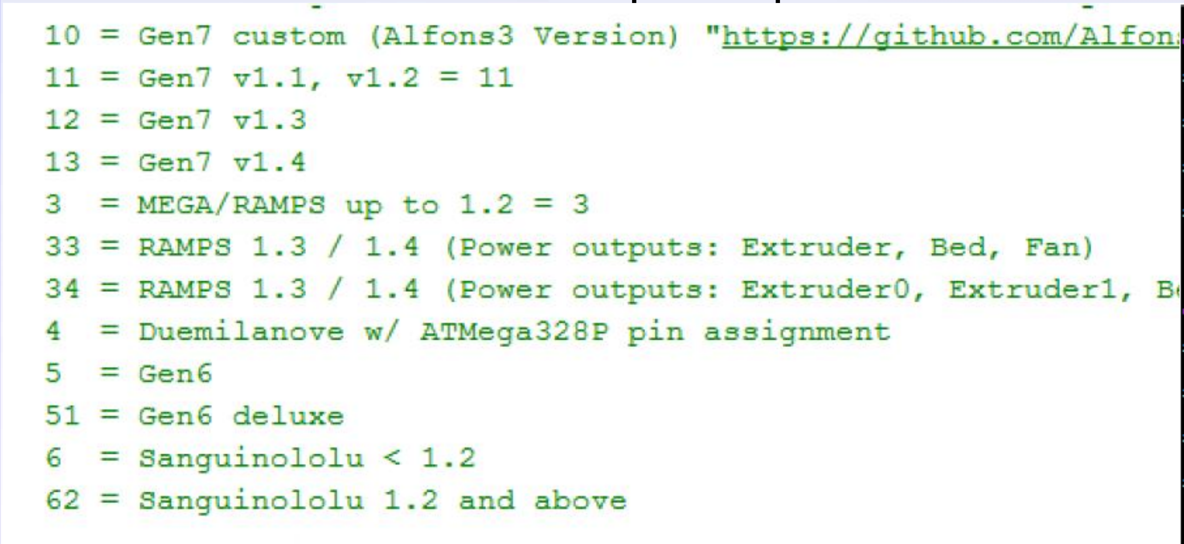
Supported board





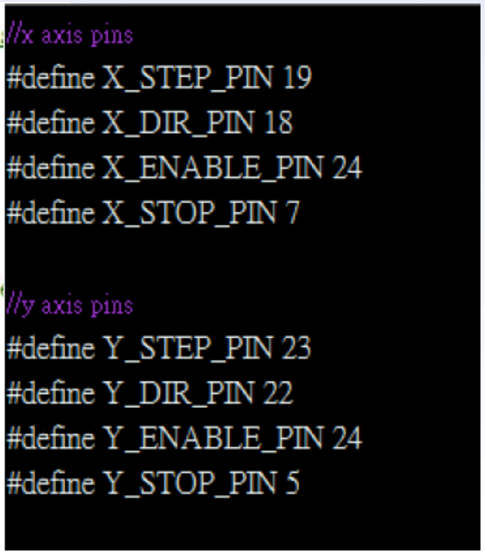
```
#ifndef MOTHERBOARD
#define MOTHERBOARD 99
#endif
```

Different board has different pin map



```
10 = Gen7 custom (Alfons3 Version) "https://github.com/Alfon
11 = Gen7 v1.1, v1.2 = 11
12 = Gen7 v1.3
13 = Gen7 v1.4
3  = MEGA/RAMPS up to 1.2 = 3
33 = RAMPS 1.3 / 1.4 (Power outputs: Extruder, Bed, Fan)
34 = RAMPS 1.3 / 1.4 (Power outputs: Extruder0, Extruder1, Be
4  = Duemilanove w/ ATMega328P pin assignment
5  = Gen6
51 = Gen6 deluxe
6  = Sanguinololu < 1.2
62 = Sanguinololu 1.2 and above
```

pins.h



```
//x axis pins
#define X_STEP_PIN 19
#define X_DIR_PIN 18
#define X_ENABLE_PIN 24
#define X_STOP_PIN 7

//y axis pins
#define Y_STEP_PIN 23
#define Y_DIR_PIN 22
#define Y_ENABLE_PIN 24
#define Y_STOP_PIN 5
```

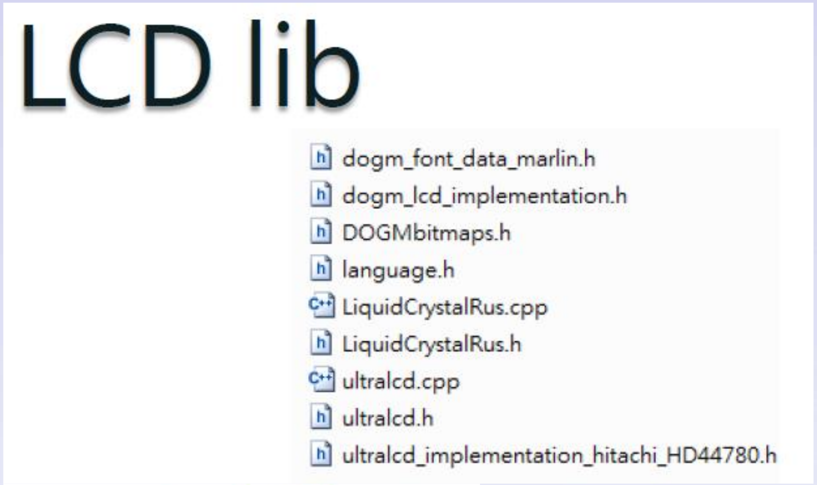
UNIVERSITY OF ARKANSAS

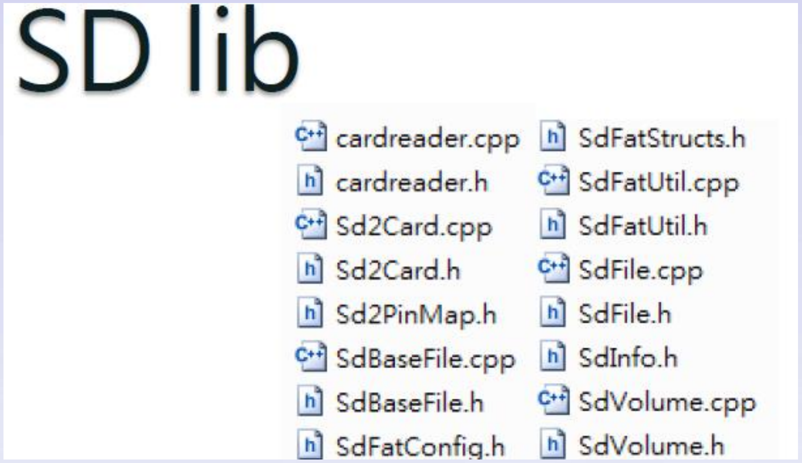
AM³ Lab
Advanced Manufacturing | Modeling | Materials

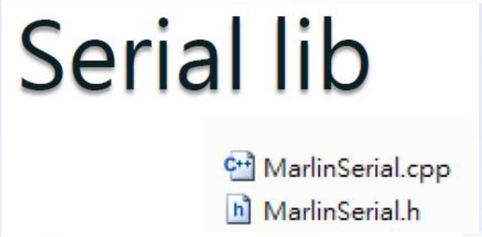
Software

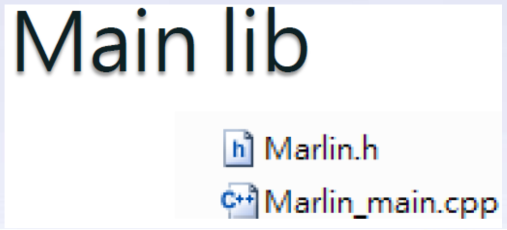
- Introduction
- What
- Why
- How
- Objectives
- Syllabus

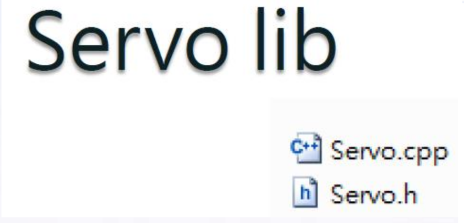
Firmware

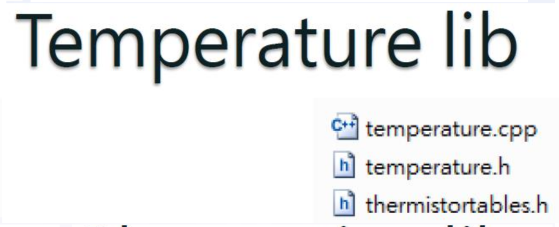

LCD lib
- dogm_font_data_marlin.h
- dogm_lcd_implementation.h
- DOGMbitmaps.h
- language.h
- LiquidCrystalRus.cpp
- LiquidCrystalRus.h
- ultralcd.cpp
- ultralcd.h
- ultralcd_implementation_hitachi_HD44780.h

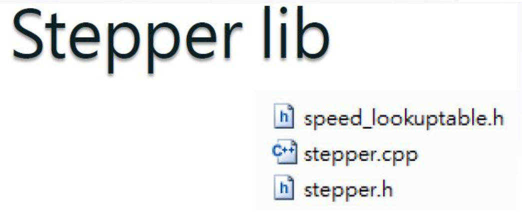

SD lib
- cardreader.cpp
- cardreader.h
- Sd2Card.cpp
- Sd2Card.h
- Sd2PinMap.h
- SdBaseFile.cpp
- SdBaseFile.h
- SdFatConfig.h
- SdFatStructs.h
- SdFatUtil.cpp
- SdFatUtil.h
- SdFile.cpp
- SdFile.h
- SdInfo.h
- SdVolume.cpp
- SdVolume.h

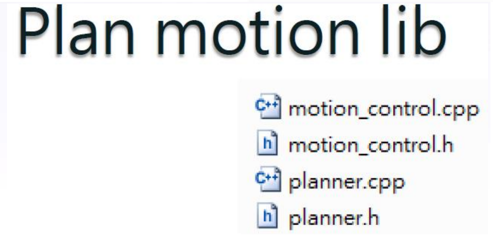

Serial lib
- MarlinSerial.cpp
- MarlinSerial.h


Main lib
- Marlin.h
- Marlin_main.cpp


Servo lib
- Servo.cpp
- Servo.h


Temperature lib
- temperature.cpp
- temperature.h
- thermistortables.h

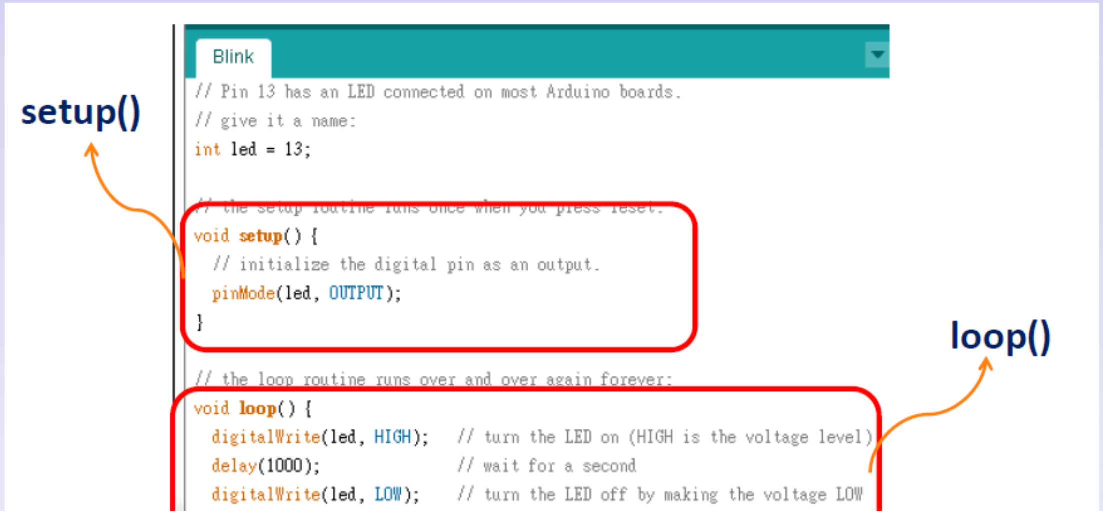

Stepper lib
- speed_lookuptable.h
- stepper.cpp
- stepper.h


Plan motion lib
- motion_control.cpp
- motion_control.h
- planner.cpp
- planner.h

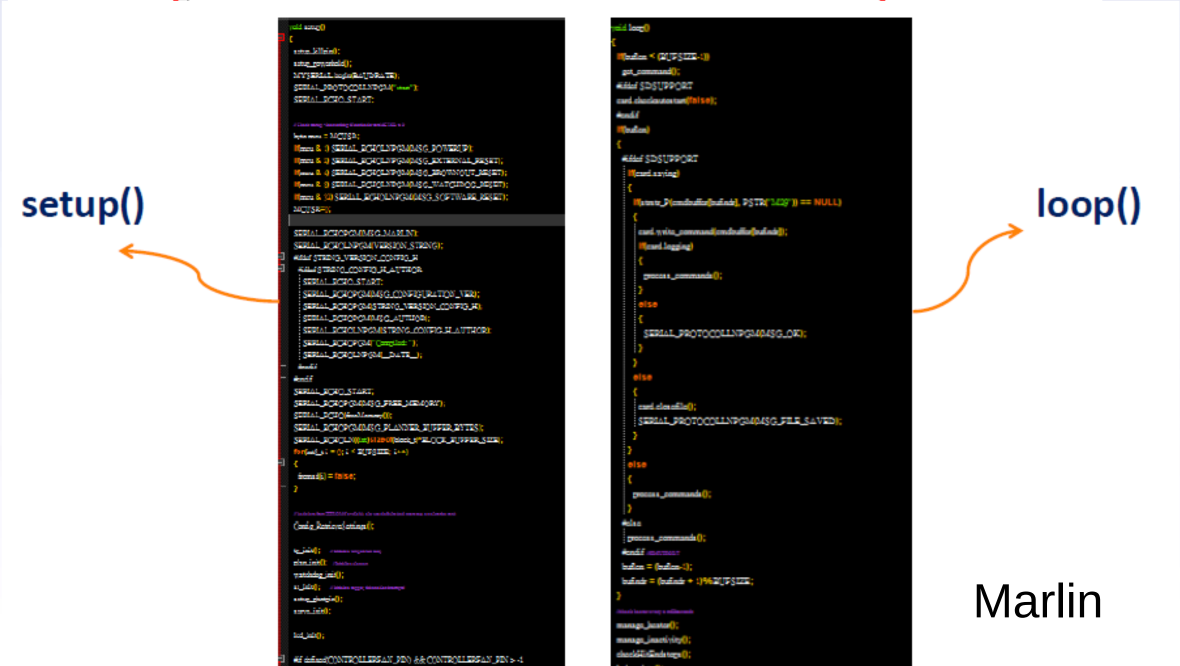
G-code for SD card
- M20 - List SD card
- M21 - Init SD card
- M22 - Release SD card
- M23 - Select SD file
- M24 - Start/resume SD print
- M25 - Pause SD print
- M26 - Set SD position in bytes
- M27 - Report SD print status
- M28 - Start SD write
- M29 - Stop SD write
- M30 - Delete file from SD

Software

Firmware – Main loop



```
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
```



```
void setup()
{
  setup_killpin();
  setup_powerhold();
  MYSERIAL.begin(BAUDRATE);
  SERIAL_PROTOCOLLNPGM("start");
  SERIAL_ECHO_START;

  // [illegible]
  byte mcu = MCUSR;
  if(mcu & 1) SERIAL_ECHOLNPGM(MSG_POWERUP);
  if(mcu & 2) SERIAL_ECHOLNPGM(MSG_EXTERNAL_RESET);
  if(mcu & 4) SERIAL_ECHOLNPGM(MSG_BROWNOUT_RESET);
  if(mcu & 8) SERIAL_ECHOLNPGM(MSG_WATCHDOG_RESET);
  if(mcu & 32) SERIAL_ECHOLNPGM(MSG_SOFTWARE_RESET);
  MCUSR=0;

  SERIAL_ECHOPGM(MSG_MARLIN);
  SERIAL_ECHOLNPGM(VERSION_STRING);
  #ifdef STRING_VERSION_CONFIG_H
   #ifdef STRING_CONFIG_H_AUTHOR
    SERIAL_ECHO_START;
    SERIAL_ECHOPGM(MSG_CONFIGURATION_VER);
    SERIAL_ECHOPGM(STRING_VERSION_CONFIG_H);
    SERIAL_ECHOPGM(MSG_AUTHOR);
    SERIAL_ECHOLNPGM(STRING_CONFIG_H_AUTHOR);
    SERIAL_ECHOPGM("Compiled: ");
    SERIAL_ECHOLNPGM(__DATE__);
   #endif
  #endif
  SERIAL_ECHO_START;
  SERIAL_ECHOPGM(MSG_FREE_MEMORY);
  SERIAL_ECHO(freeMemory());
  SERIAL_ECHOPGM(MSG_PLANNER_BUFFER_BYTES);
  SERIAL_ECHOLN((int)sizeof(block_t)*BLOCK_BUFFER_SIZE);
  for(int8_t i = 0; i < BUFSIZE; i++)
  {
   fromsd[i] = false;
  }

  // [illegible]
  Config_RetrieveSettings();

  tp_init();    // [illegible]
  plan_init();  // [illegible]
  watchdog_init();
  st_init();    // [illegible]
  setup_photpin();
  servo_init();

  lcd_init();

  #if defined(CONTROLLERFAN_PIN) && CONTROLLERFAN_PIN > -1
```

```
void loop()
{
  if(buflen < (BUFSIZE-1))
   get_command();
  #ifdef SDSUPPORT
  card.checkautostart(false);
  #endif
  if(buflen)
  {
   #ifdef SDSUPPORT
    if(card.saving)
    {
     if(strstr_P(cmdbuffer[bufindr], PSTR("M29")) == NULL)
     {
      card.write_command(cmdbuffer[bufindr]);
      if(card.logging)
      {
       process_commands();
      }
      else
      {
       SERIAL_PROTOCOLLNPGM(MSG_OK);
      }
     }
     else
     {
      card.closefile();
      SERIAL_PROTOCOLLNPGM(MSG_FILE_SAVED);
     }
    }
    else
    {
     process_commands();
    }
   #else
    process_commands();
   #endif //SDSUPPORT
   buflen = (buflen-1);
   bufindr = (bufindr + 1)%BUFSIZE;
  }
  //[illegible]
  manage_heater();
  manage_inactivity();
  checkHitEndstops();
```

Marlin

Software

Introduction

What

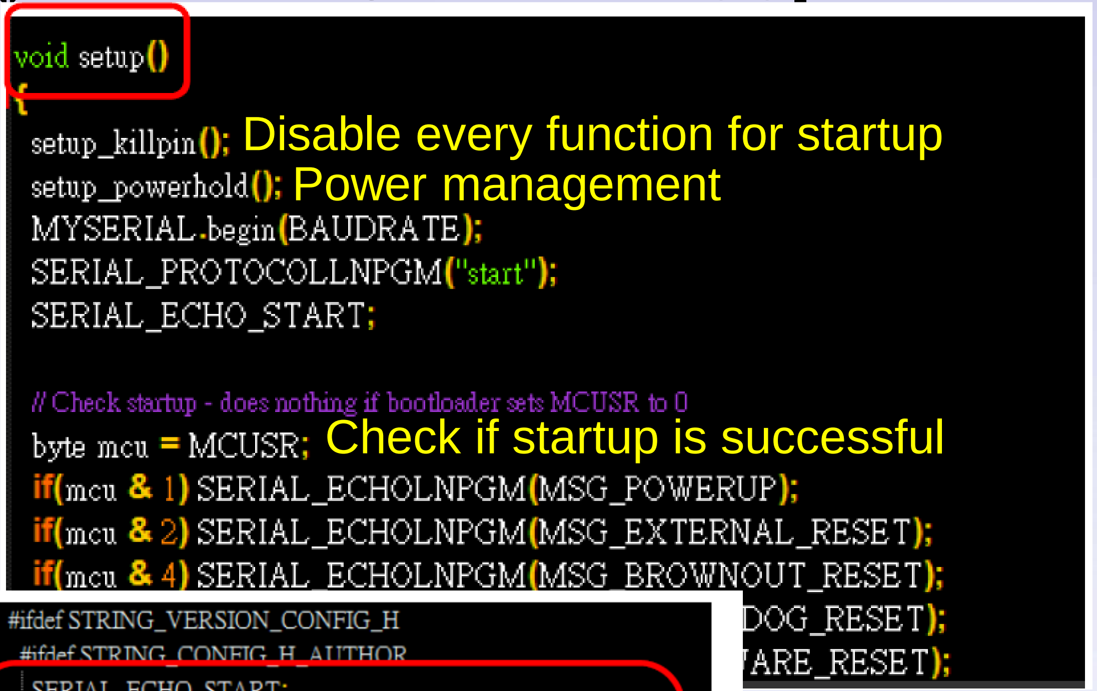
Why

How

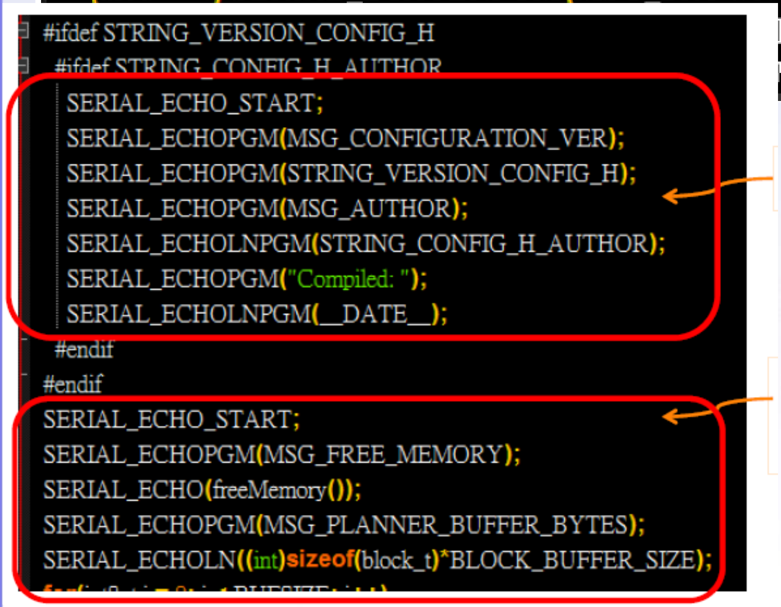
Objectives

Syllabus

Firmware – Main loop



```
void setup()
{
  setup_killpin(); Disable every function for startup
  setup_powerhold(); Power management
  MYSERIAL.begin(BAUDRATE);
  SERIAL_PROTOCOLLNPGM("start");
  SERIAL_ECHO_START;

  // Check startup - does nothing if bootloader sets MCUSR to 0
  byte mcu = MCUSR; Check if startup is successful
  if(mcu & 1) SERIAL_ECHOLNPGM(MSG_POWERUP);
  if(mcu & 2) SERIAL_ECHOLNPGM(MSG_EXTERNAL_RESET);
  if(mcu & 4) SERIAL_ECHOLNPGM(MSG_BROWNOUT_RESET);
                                   DOG_RESET);
                                   ARE_RESET);
```



```
#ifdef STRING_VERSION_CONFIG_H
  #ifdef STRING_CONFIG_H_AUTHOR
    SERIAL_ECHO_START;
    SERIAL_ECHOPGM(MSG_CONFIGURATION_VER);
    SERIAL_ECHOPGM(STRING_VERSION_CONFIG_H);
    SERIAL_ECHOPGM(MSG_AUTHOR);
    SERIAL_ECHOLNPGM(STRING_CONFIG_H_AUTHOR);
    SERIAL_ECHOPGM("Compiled: ");
    SERIAL_ECHOLNPGM(__DATE__);
  #endif
#endif
SERIAL_ECHO_START;
SERIAL_ECHOPGM(MSG_FREE_MEMORY);
SERIAL_ECHO(freeMemory());
SERIAL_ECHOPGM(MSG_PLANNER_BUFFER_BYTES);
SERIAL_ECHOLN((int)sizeof(block_t)*BLOCK_BUFFER_SIZE);
```

Version info

Memory info

Software

Introduction

What

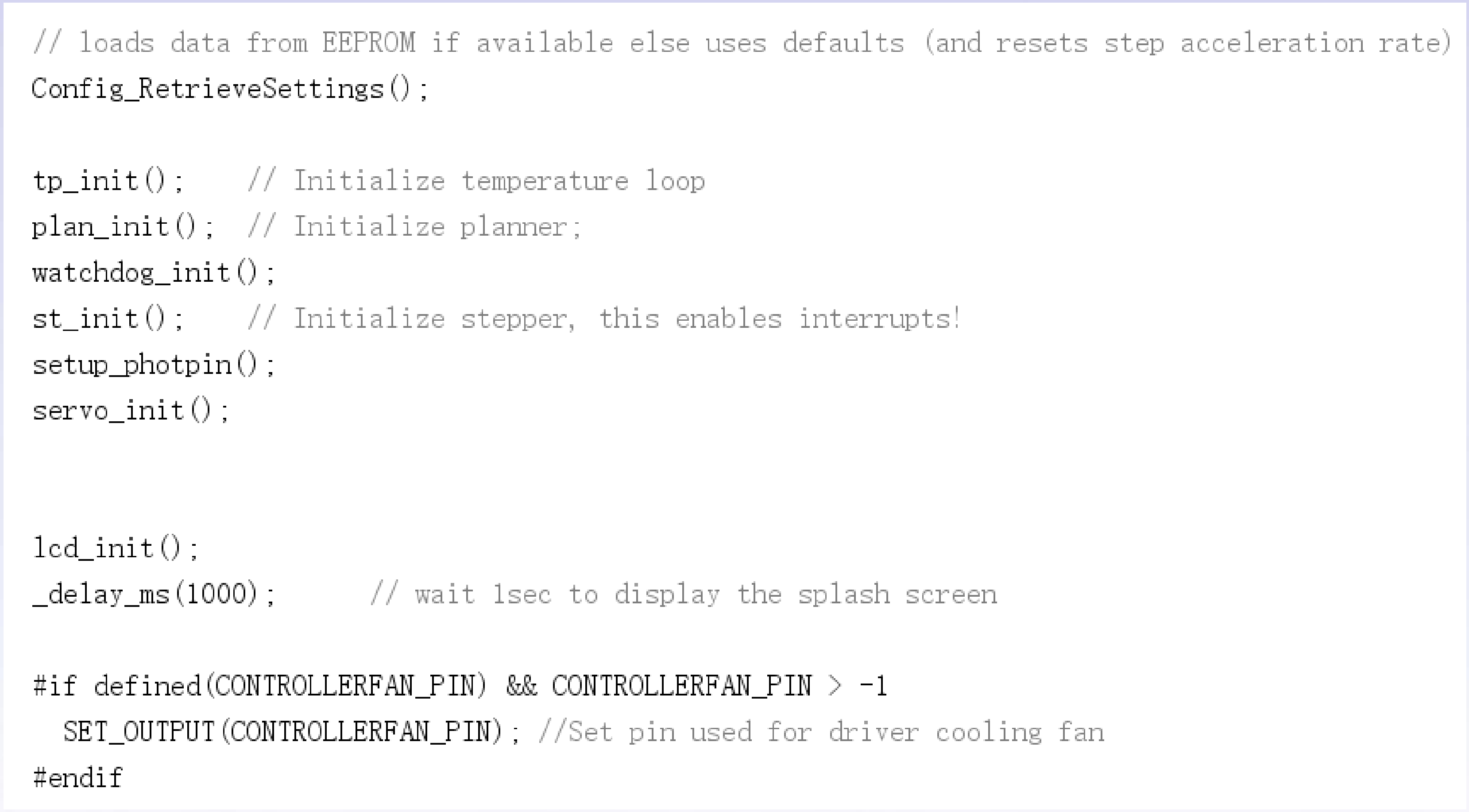
Why

How

Objectives

Syllabus

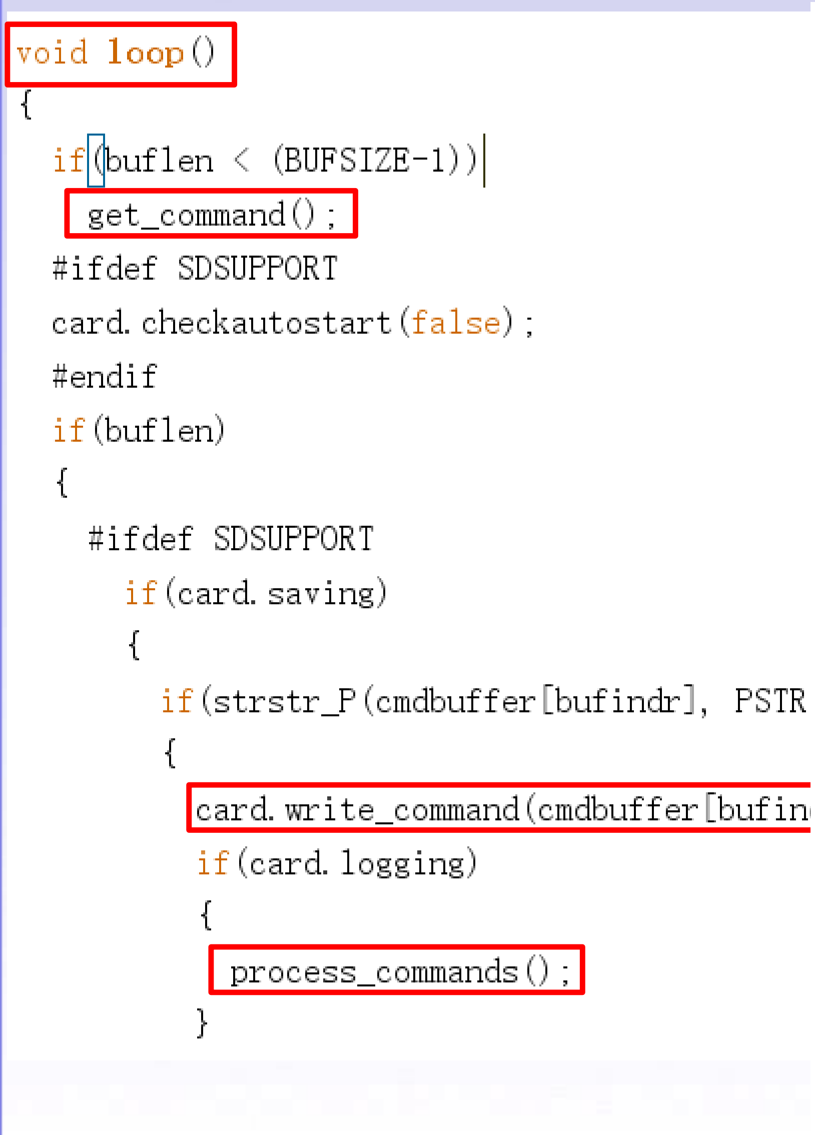
Firmware – Main loop



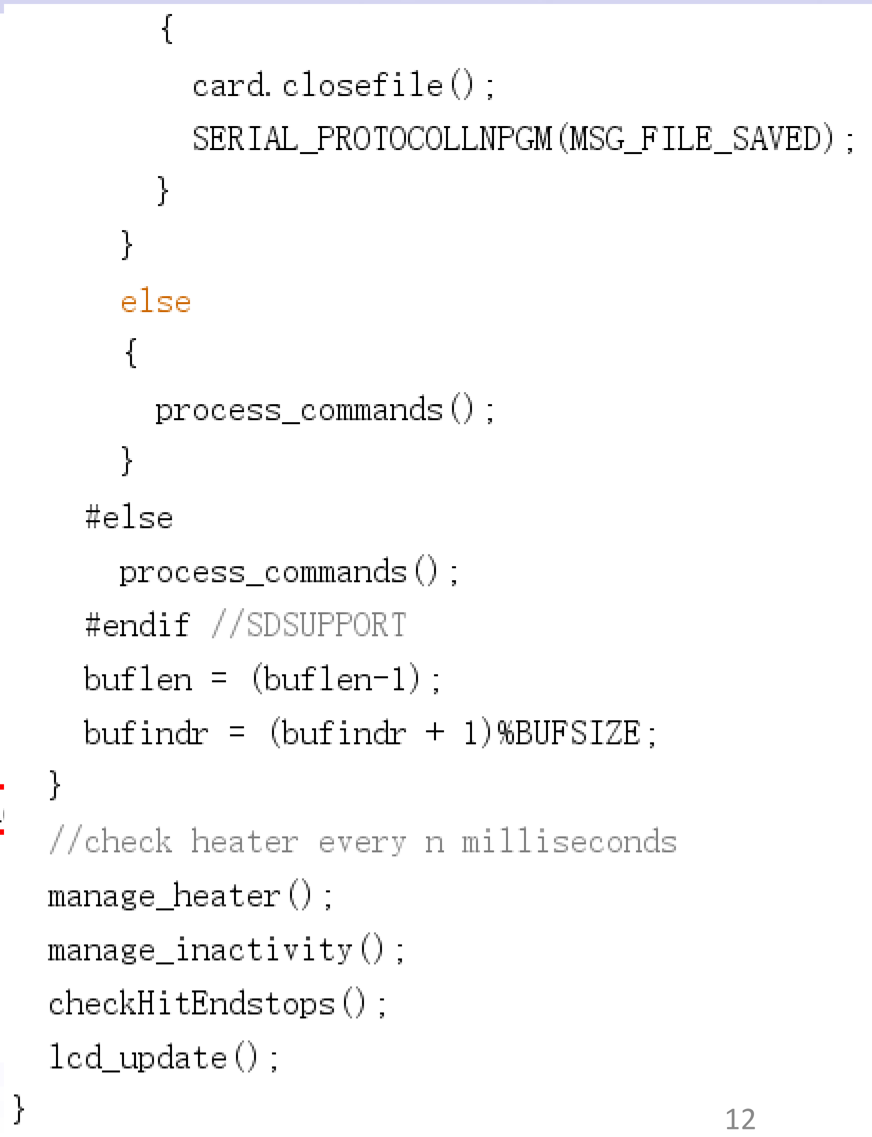
```
// loads data from EEPROM if available else uses defaults (and resets step acceleration rate)
Config_RetrieveSettings();

tp_init();    // Initialize temperature loop
plan_init();  // Initialize planner;
watchdog_init();
st_init();    // Initialize stepper, this enables interrupts!
setup_photpin();
servo_init();


lcd_init();
_delay_ms(1000);      // wait 1sec to display the splash screen

#if defined(CONTROLLERFAN_PIN) && CONTROLLERFAN_PIN > -1
  SET_OUTPUT(CONTROLLERFAN_PIN); //Set pin used for driver cooling fan
#endif
```

Software

Firmware – Main loop



```
void loop()
{
  if(buflen < (BUFSIZE-1))
    get_command();
  #ifdef SDSUPPORT
  card.checkautostart(false);
  #endif
  if(buflen)
  {
    #ifdef SDSUPPORT
      if(card.saving)
      {
        if(strstr_P(cmdbuffer[bufindr], PSTR
        {
          card.write_command(cmdbuffer[bufin
          if(card.logging)
          {
            process_commands();
          }
```

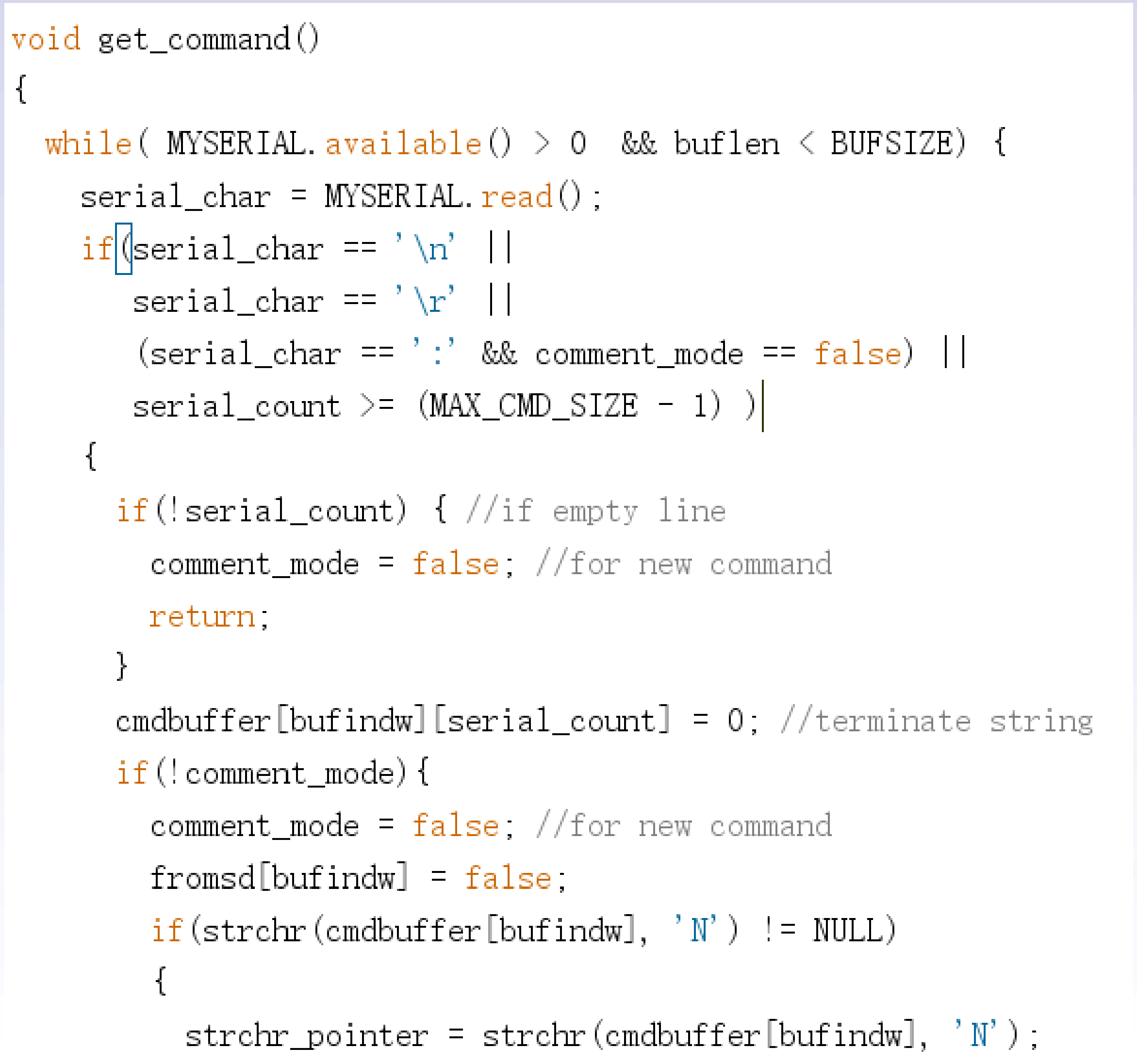


```
          {
            card.closefile();
            SERIAL_PROTOCOLLNPGM(MSG_FILE_SAVED);
          }
        }
        else
        {
          process_commands();
        }
      #else
        process_commands();
      #endif //SDSUPPORT
      buflen = (buflen-1);
      bufindr = (bufindr + 1)%BUFSIZE;
    }
    //check heater every n milliseconds
    manage_heater();
    manage_inactivity();
    checkHitEndstops();
    lcd_update();
}
```

Software

Firmware – Main loop

get_command(): Getting g-code from serial port



```
void get_command()
{
  while( MYSERIAL.available() > 0  && buflen < BUFSIZE) {
    serial_char = MYSERIAL.read();
    if(serial_char == '\n' ||
       serial_char == '\r' ||
       (serial_char == ':' && comment_mode == false) ||
       serial_count >= (MAX_CMD_SIZE - 1) )
    {
      if(!serial_count) { //if empty line
        comment_mode = false; //for new command
        return;
      }
      cmdbuffer[bufindw][serial_count] = 0; //terminate string
      if(!comment_mode){
        comment_mode = false; //for new command
        fromsd[bufindw] = false;
        if(strchr(cmdbuffer[bufindw], 'N') != NULL)
        {
          strchr_pointer = strchr(cmdbuffer[bufindw], 'N');
```

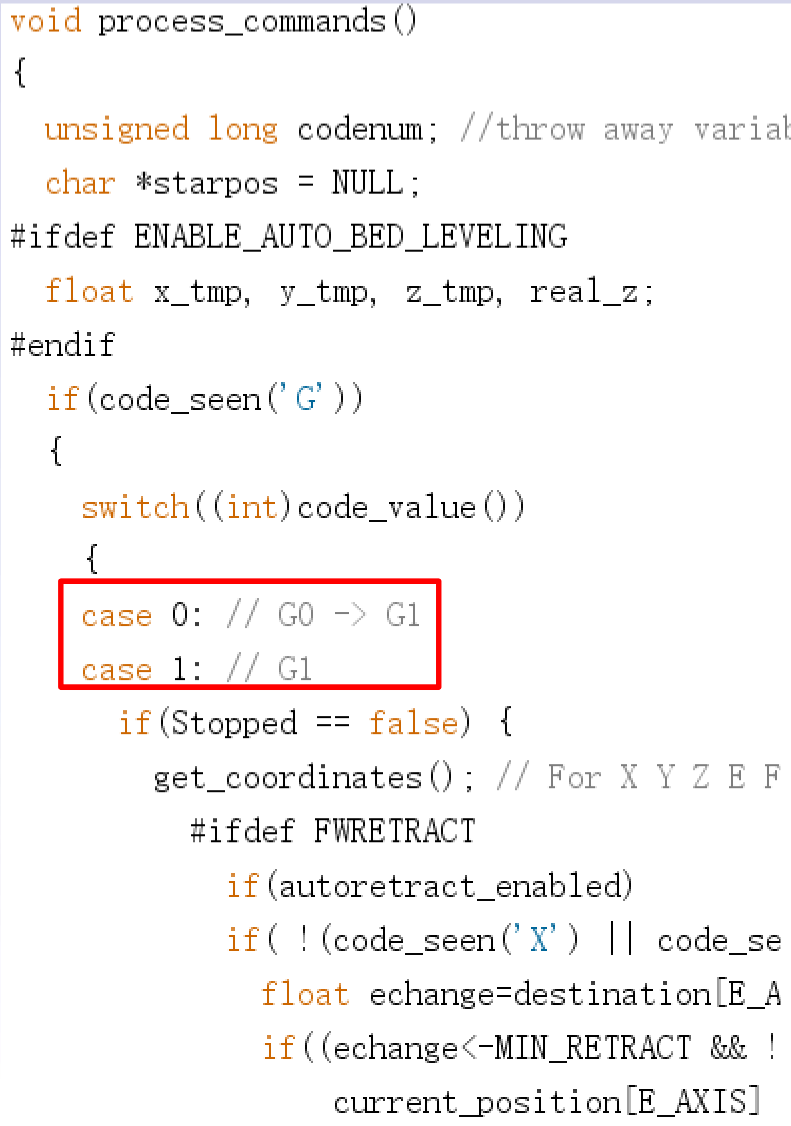
UNIVERSITY OF ARKANSAS

Software

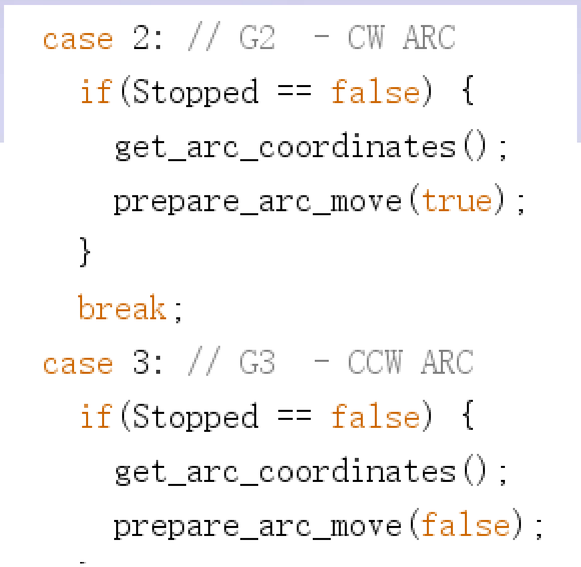
AM³ Lab
Advanced Manufacturing | Modeling | Materials

Firmware – Main loop

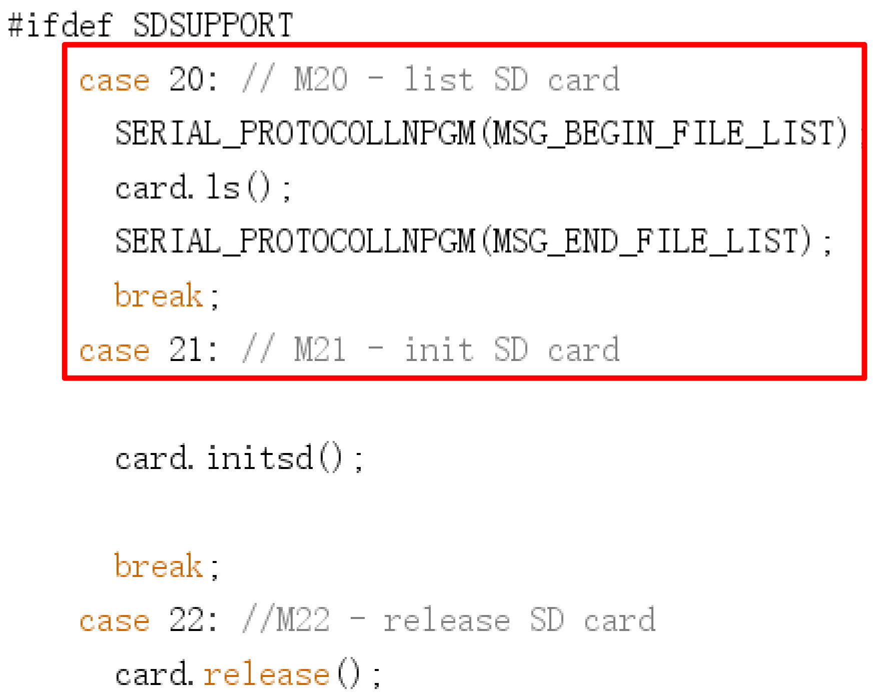
process_commands(): process g-code



```
void process_commands()
{
  unsigned long codenum; //throw away variable
  char *starpos = NULL;
#ifdef ENABLE_AUTO_BED_LEVELING
  float x_tmp, y_tmp, z_tmp, real_z;
#endif
  if(code_seen('G'))
  {
    switch((int)code_value())
    {
    case 0: // G0 -> G1
    case 1: // G1
      if(Stopped == false) {
        get_coordinates(); // For X Y Z E F
          #ifdef FWRETRACT
            if(autoretract_enabled)
            if( !(code_seen('X') || code_se
              float echange=destination[E_A
              if((echange<-MIN_RETRACT && !
                  current_position[E_AXIS]
```



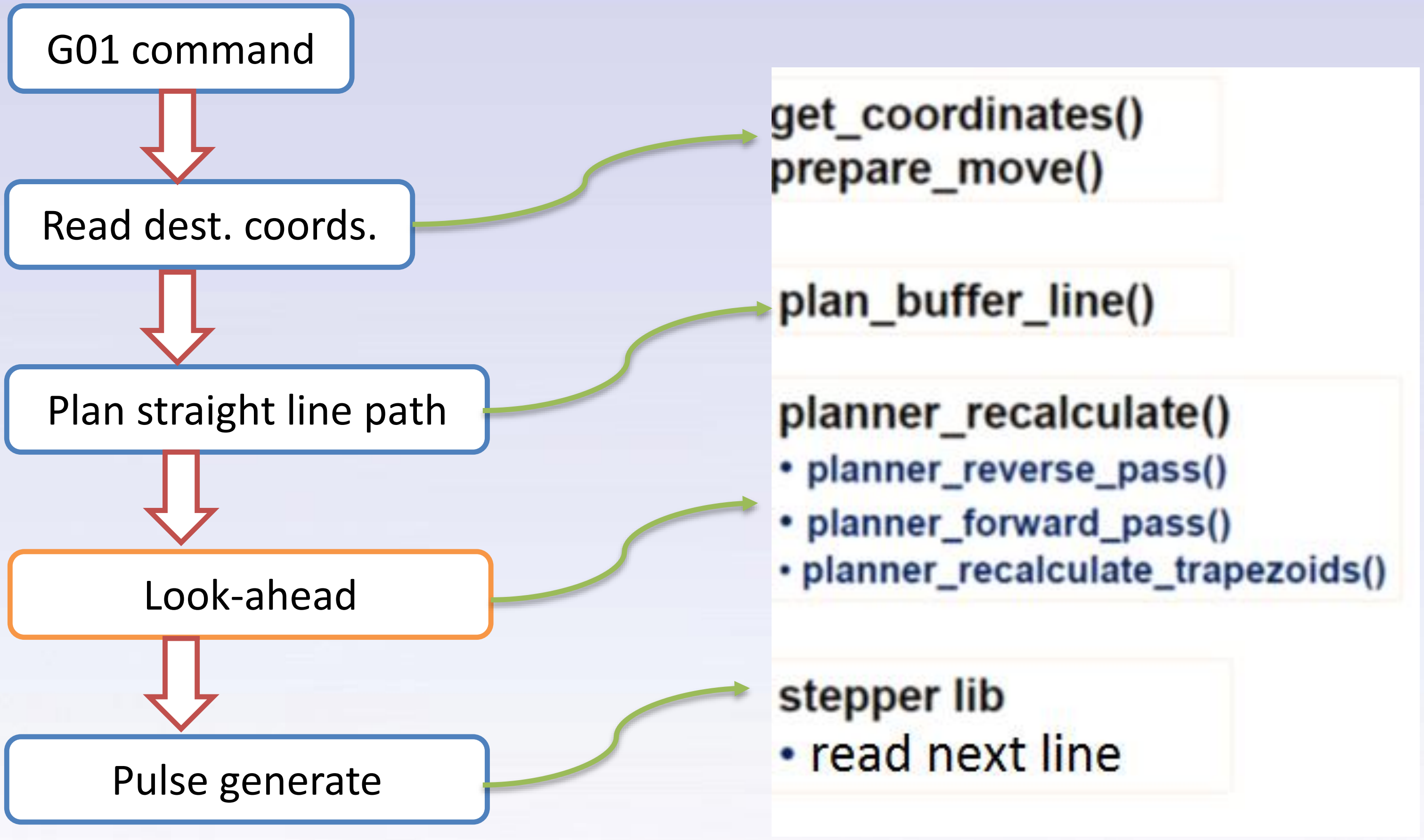
```
case 2: // G2  - CW ARC
  if(Stopped == false) {
    get_arc_coordinates();
    prepare_arc_move(true);
  }
  break;
case 3: // G3  - CCW ARC
  if(Stopped == false) {
    get_arc_coordinates();
    prepare_arc_move(false);
```



```
#ifdef SDSUPPORT
    case 20: // M20 - list SD card
      SERIAL_PROTOCOLLNPGM(MSG_BEGIN_FILE_LIST)
      card.ls();
      SERIAL_PROTOCOLLNPGM(MSG_END_FILE_LIST);
      break;
    case 21: // M21 - init SD card

      card.initsd();

      break;
    case 22: //M22 - release SD card
      card.release();
```

Software

Firmware – Motion plan



Look-ahead will only decelerate and accelerate to some non-zero velocity, but not completely stop

UNIVERSITY OF ARKANSAS

Software

AM³ Lab
Advanced Manufacturing | Modeling | Materials

Introduction

What

Why

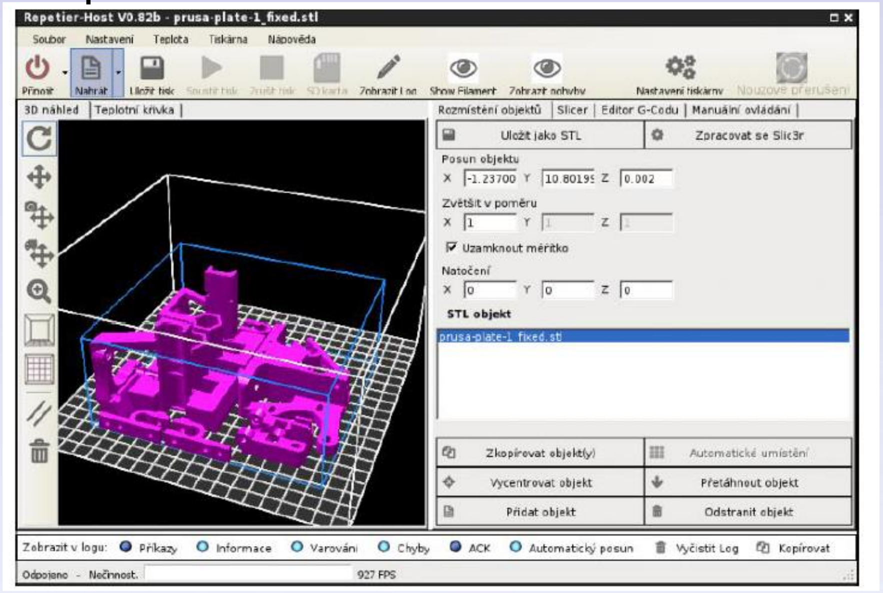
How

Objectives

Syllabus

Host software – send g-code

Repetier host



Posun objektu X -1.23700 Y 10.8019 Z 0.002

Zvětšit v poměru X 1 Y 1 Z 1

☑ Uzamknout měřítko

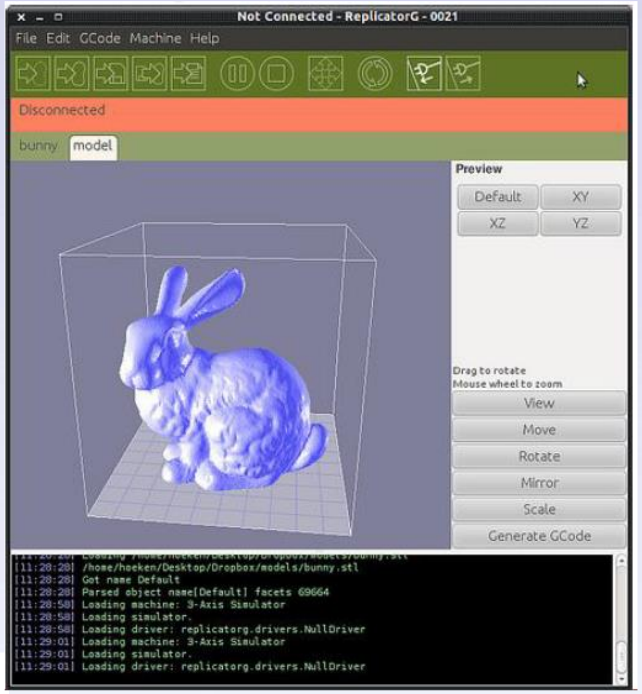
Natočení X 0 Y 0 Z 0

STL objekt

prusa-plate-1_fixed.stl

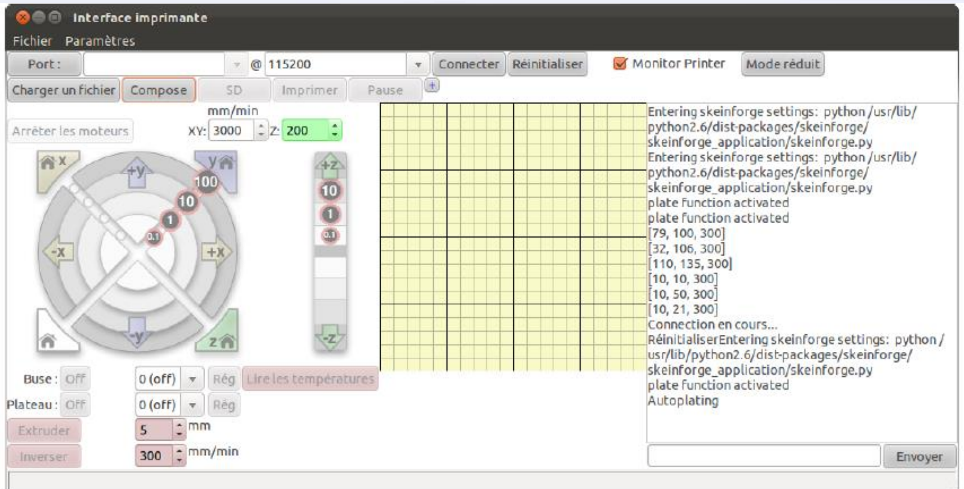
Odpojeno - Nečinnost. 927 FPS

ReplicatorG



```
[11:28:28] /home/hoeken/Desktop/Dropbox/models/bunny.stl
[11:28:28] Got name Default
[11:28:28] Parsed object name[Default] facets 69664
[11:28:58] Loading machine: 3-Axis Simulator
[11:28:58] Loading simulator.
[11:28:58] Loading driver: replicatorg.drivers.NullDriver
[11:29:01] Loading machine: 3-Axis Simulator
[11:29:01] Loading simulator.
[11:29:01] Loading driver: replicatorg.drivers.NullDriver
```

Printrun



@ 115200

☑ Monitor Printer

mm/min XY: 3000 Z: 200

Buse : Off 0 (off) Rég

Plateau : Off 0 (off) Rég

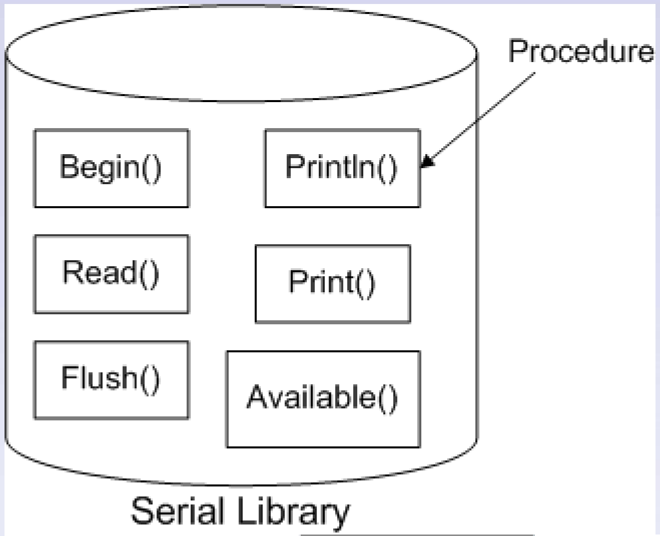
Extruder 5 mm

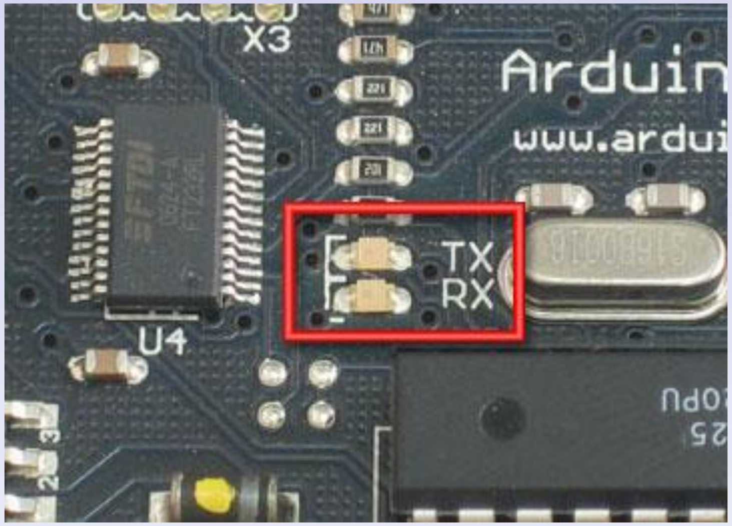
Inverser 300 mm/min

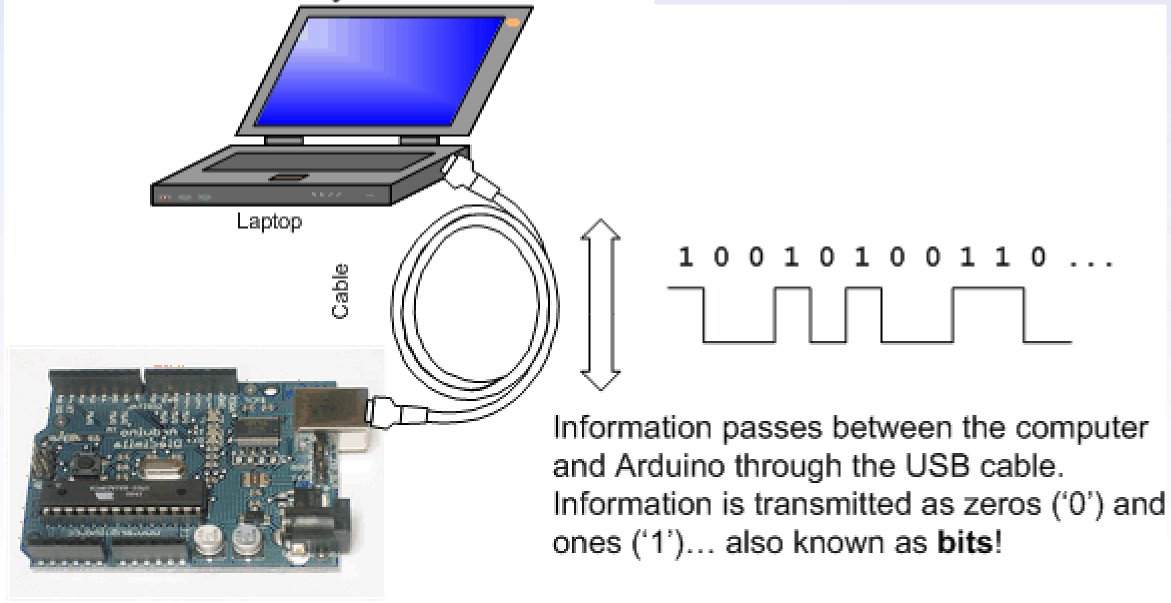
```
Entering skeinforge settings:  python /usr/lib/
python2.6/dist-packages/skeinforge/
skeinforge_application/skeinforge.py
Entering skeinforge settings:  python /usr/lib/
python2.6/dist-packages/skeinforge/
skeinforge_application/skeinforge.py
plate function activated
plate function activated
[79, 100, 300]
[32, 106, 300]
[110, 135, 300]
[10, 10, 300]
[10, 50, 300]
[10, 21, 300]
Connection en cours...
RéinitialiserEntering skeinforge settings:  python /
usr/lib/python2.6/dist-packages/skeinforge/
skeinforge_application/skeinforge.py
plate function activated
Autoplating
```

Software

Host software – serial communication





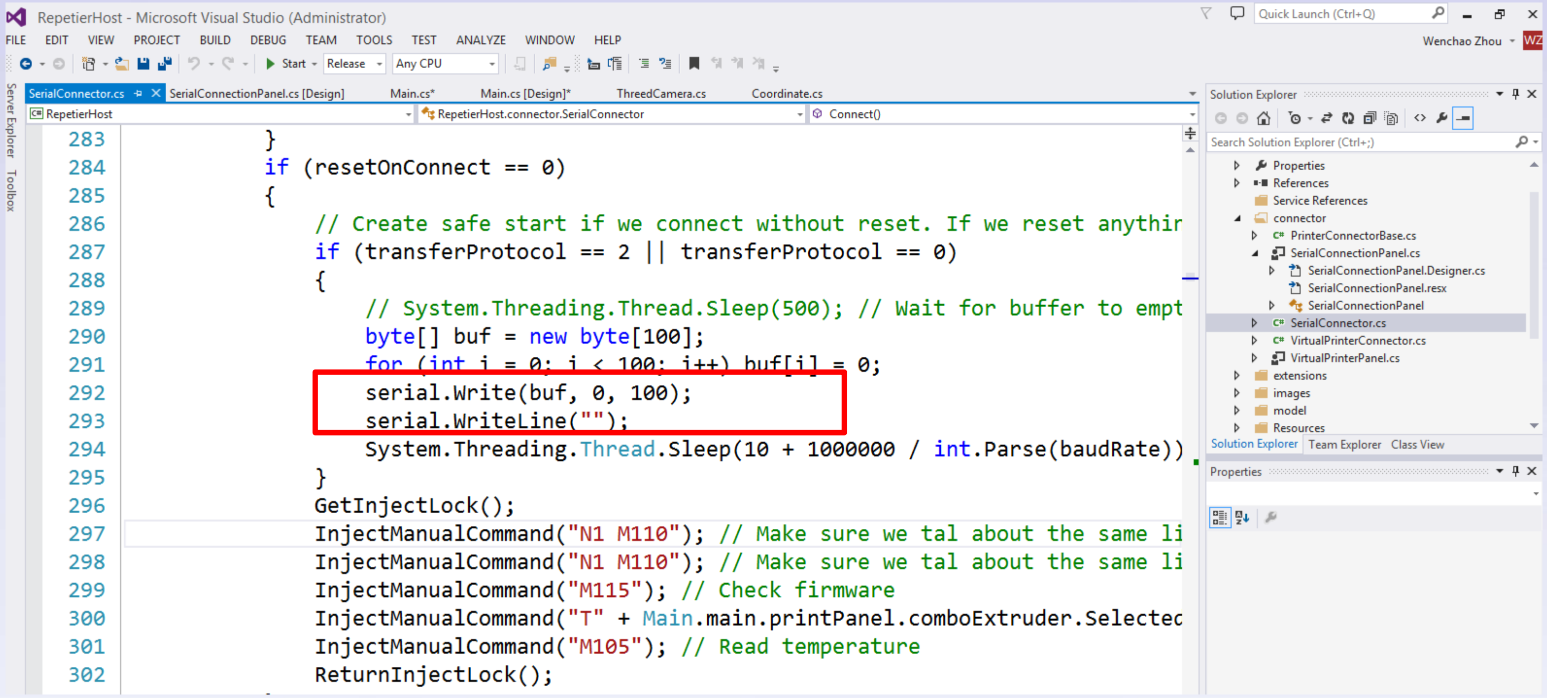


Information passes between the computer and Arduino through the USB cable. Information is transmitted as zeros ('0') and ones ('1')… also known as **bits**!

Software

- Introduction
- What
- Why
- How
- Objectives
- Syllabus

Host software – serial communication


RepetierHost - Microsoft Visual Studio (Administrator)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
Start Release Any CPU
SerialConnector.cs SerialConnectionPanel.cs [Design] Main.cs* Main.cs [Design]* ThreedCamera.cs Coordinate.cs
RepetierHost RepetierHost.connector.SerialConnector Connect()
Solution Explorer
Properties
References
Service References
connector
PrinterConnectorBase.cs
SerialConnectionPanel.cs
SerialConnectionPanel.Designer.cs
SerialConnectionPanel.resx
SerialConnectionPanel
SerialConnector.cs
VirtualPrinterConnector.cs
VirtualPrinterPanel.cs
extensions
images
model
Resources
Solution Explorer Team Explorer Class View
Properties

```
            }
            if (resetOnConnect == 0)
            {
                // Create safe start if we connect without reset. If we reset anythir
                if (transferProtocol == 2 || transferProtocol == 0)
                {
                    // System.Threading.Thread.Sleep(500); // Wait for buffer to empt
                    byte[] buf = new byte[100];
                    for (int i = 0; i < 100; i++) buf[i] = 0;
                    serial.Write(buf, 0, 100);
                    serial.WriteLine("");
                    System.Threading.Thread.Sleep(10 + 1000000 / int.Parse(baudRate))
                }
                GetInjectLock();
                InjectManualCommand("N1 M110"); // Make sure we tal about the same li
                InjectManualCommand("N1 M110"); // Make sure we tal about the same li
                InjectManualCommand("M115"); // Check firmware
                InjectManualCommand("T" + Main.main.printPanel.comboExtruder.Selectec
                InjectManualCommand("M105"); // Read temperature
                ReturnInjectLock();
```

Repetier-Host: Sending g-code to serial port from PC (C#)

UNIVERSITY OF ARKANSAS

Software

AM³ Lab
Advanced Manufacturing | Modeling | Materials

Introduction

What

Why

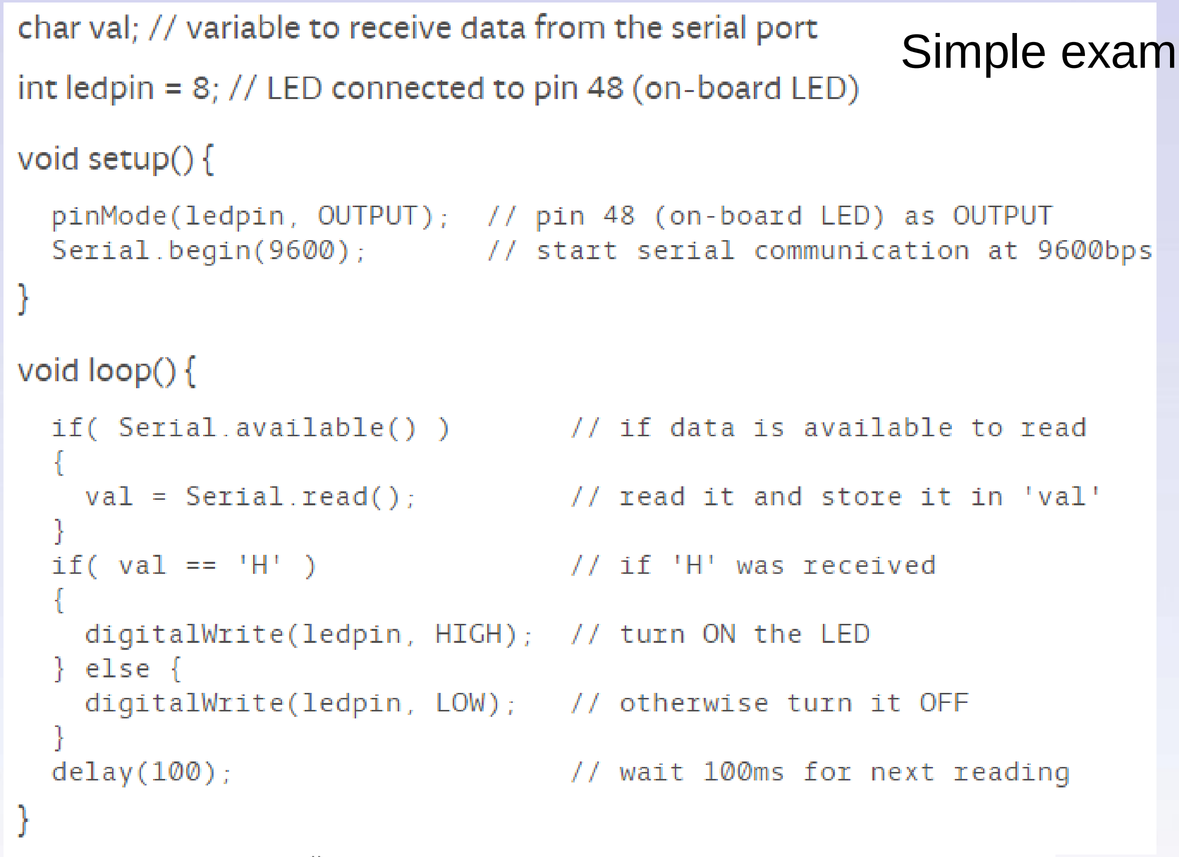
How

Objectives

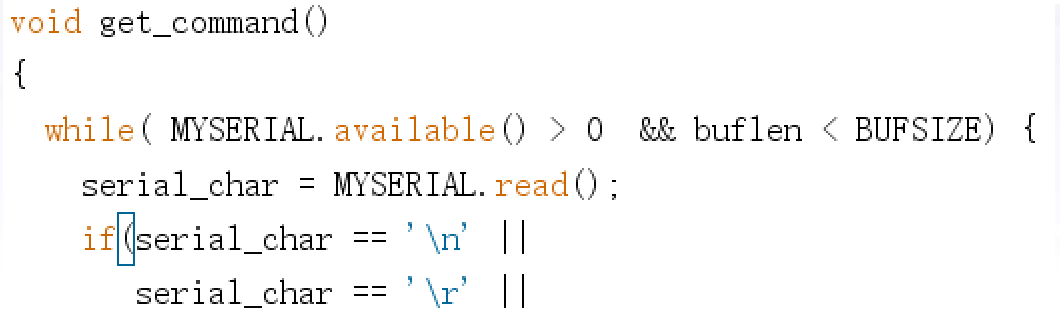
Syllabus

Host software – serial communication

Simple example of serial.read



```
char val; // variable to receive data from the serial port

int ledpin = 8; // LED connected to pin 48 (on-board LED)

void setup() {

  pinMode(ledpin, OUTPUT);  // pin 48 (on-board LED) as OUTPUT
  Serial.begin(9600);       // start serial communication at 9600bps
}

void loop() {

  if( Serial.available() )       // if data is available to read
  {
    val = Serial.read();         // read it and store it in 'val'
  }
  if( val == 'H' )               // if 'H' was received
  {
    digitalWrite(ledpin, HIGH);  // turn ON the LED
  } else {
    digitalWrite(ledpin, LOW);   // otherwise turn it OFF
  }
  delay(100);                    // wait 100ms for next reading
}
```



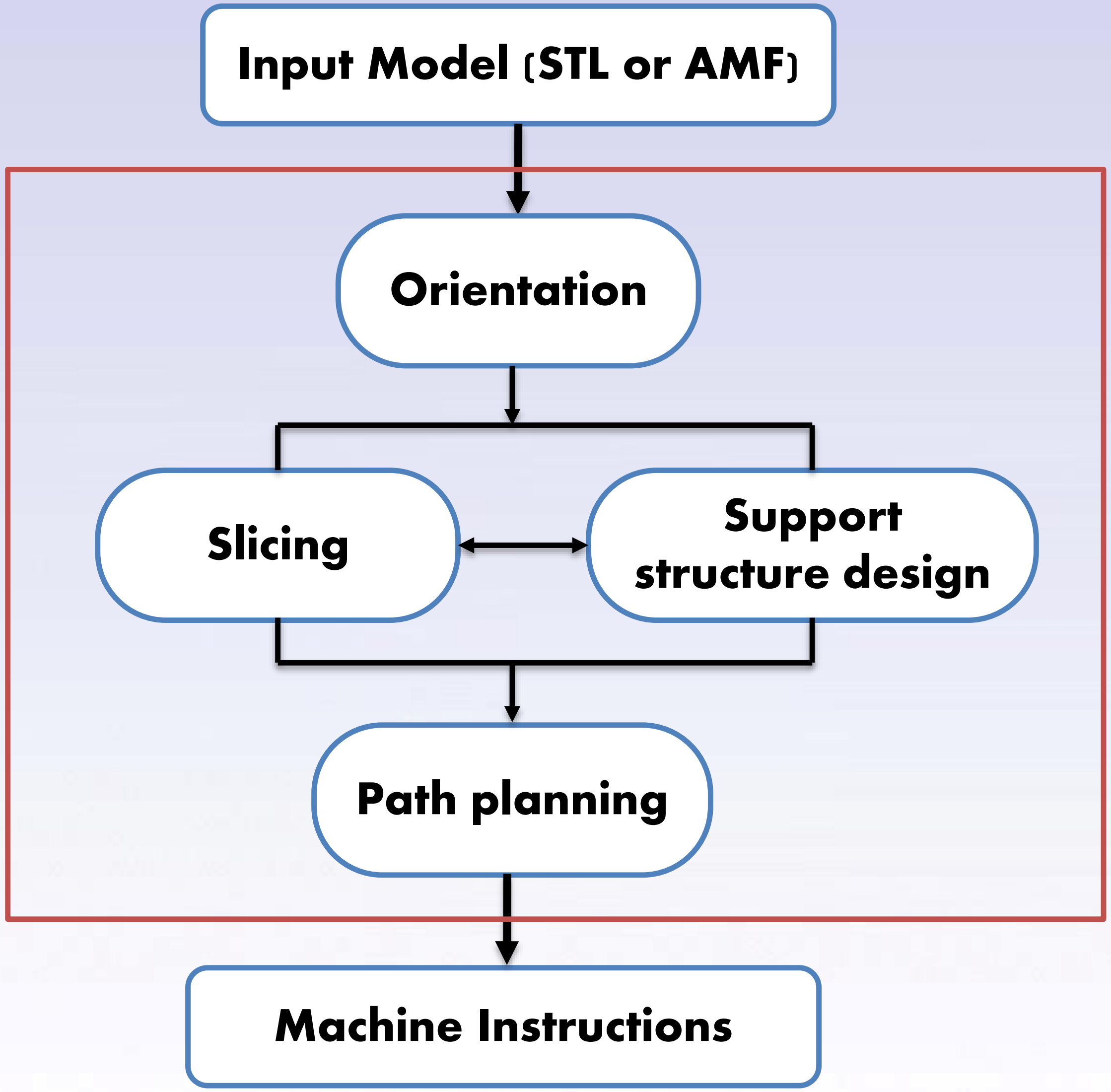
```
void get_command()
{
  while( MYSERIAL.available() > 0  && buflen < BUFSIZE) {
    serial_char = MYSERIAL.read();
    if(serial_char == '\n' ||
       serial_char == '\r' ||
```

UNIVERSITY OF ARKANSAS

AM³ Lab
Advanced Manufacturing | Modeling | Materials

Software

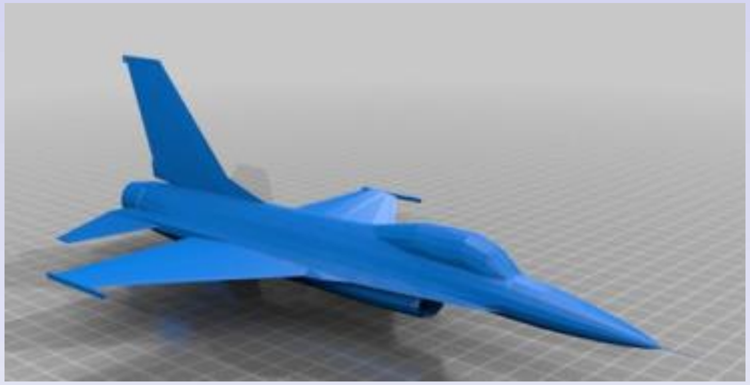
Slicing

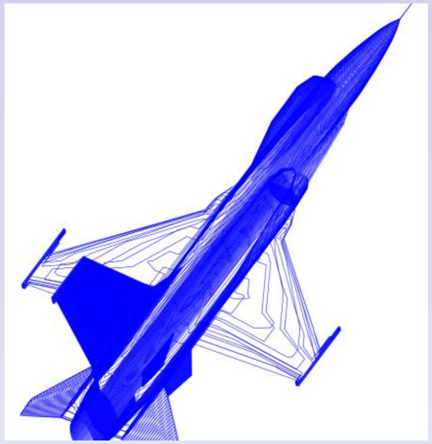


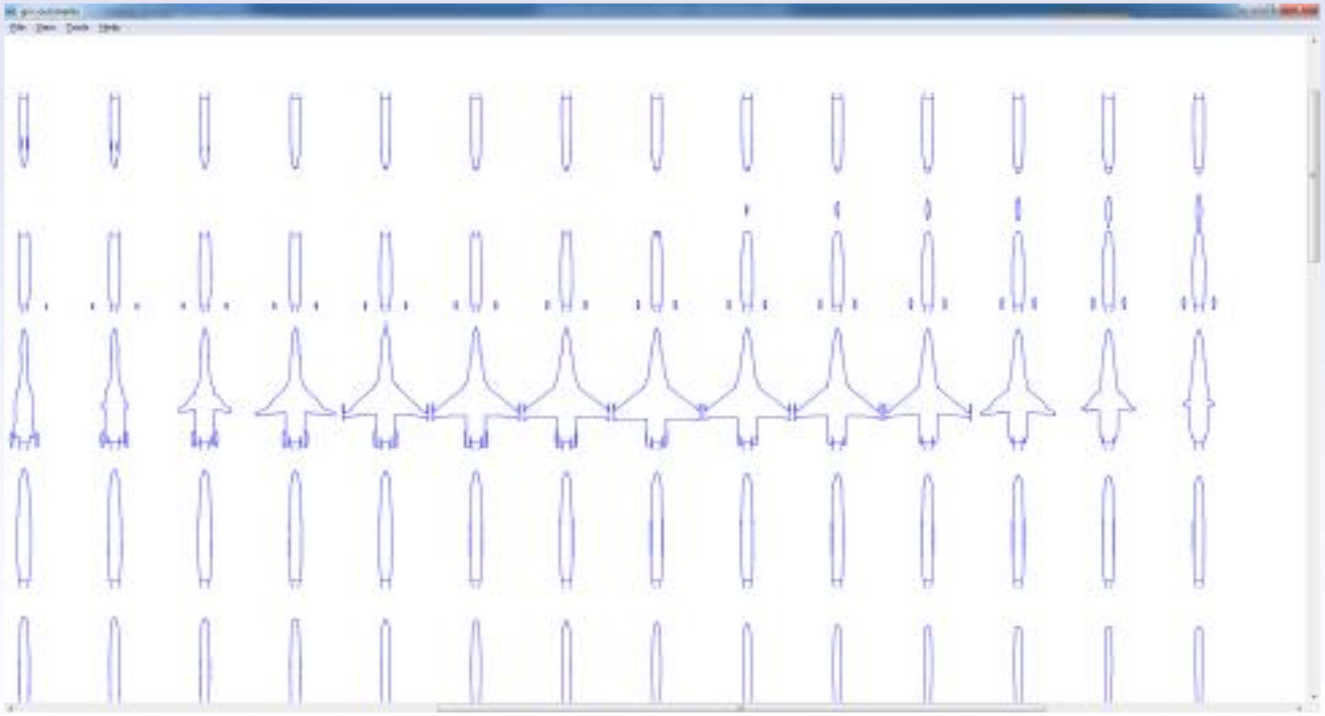
- Orientation
- Support structure
- Slicing
- Path planning
- Machine instructions

Software

Slicing







Picture credit: Raveh Gonen

Sliced results

UNIVERSITY OF ARKANSAS

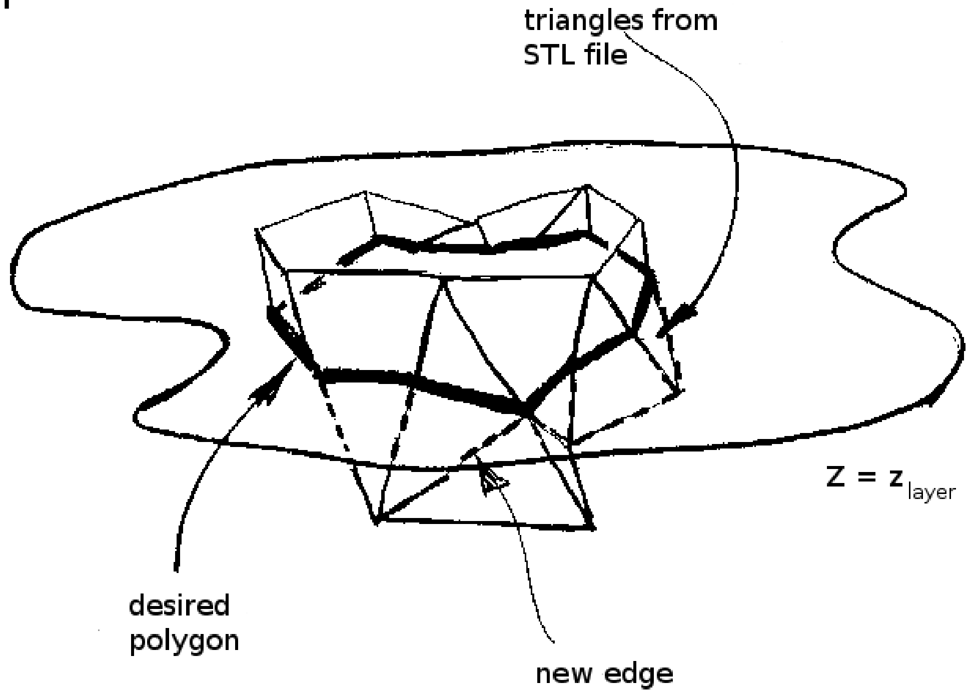
Software

AM³ Lab
Advanced Manufacturing | Modeling | Materials

Slicing – Data structure

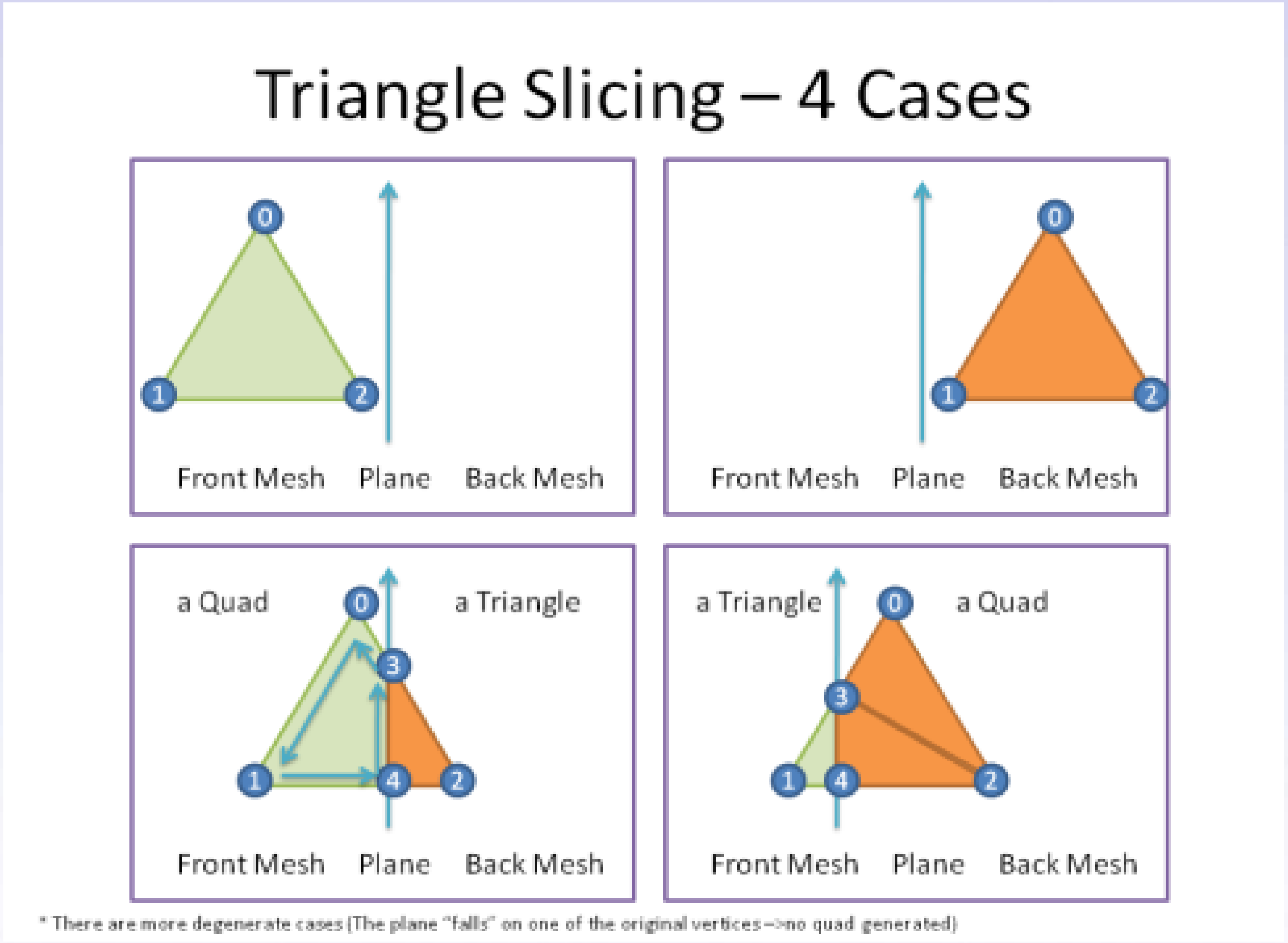
- v3 -> a vector with 3 floats {x,y,z}
- LineSegment -> {v3 point0, point1}
- Plane -> {v3 normal, float distance}
- Triangle -> {v3 vertices[3], normal}
- TriangleMesh -> {vector of Triangle}
- nSlices -> compute-number-of-slices using slice-size

Z-plane intersects model



Software

Slicing – analysis

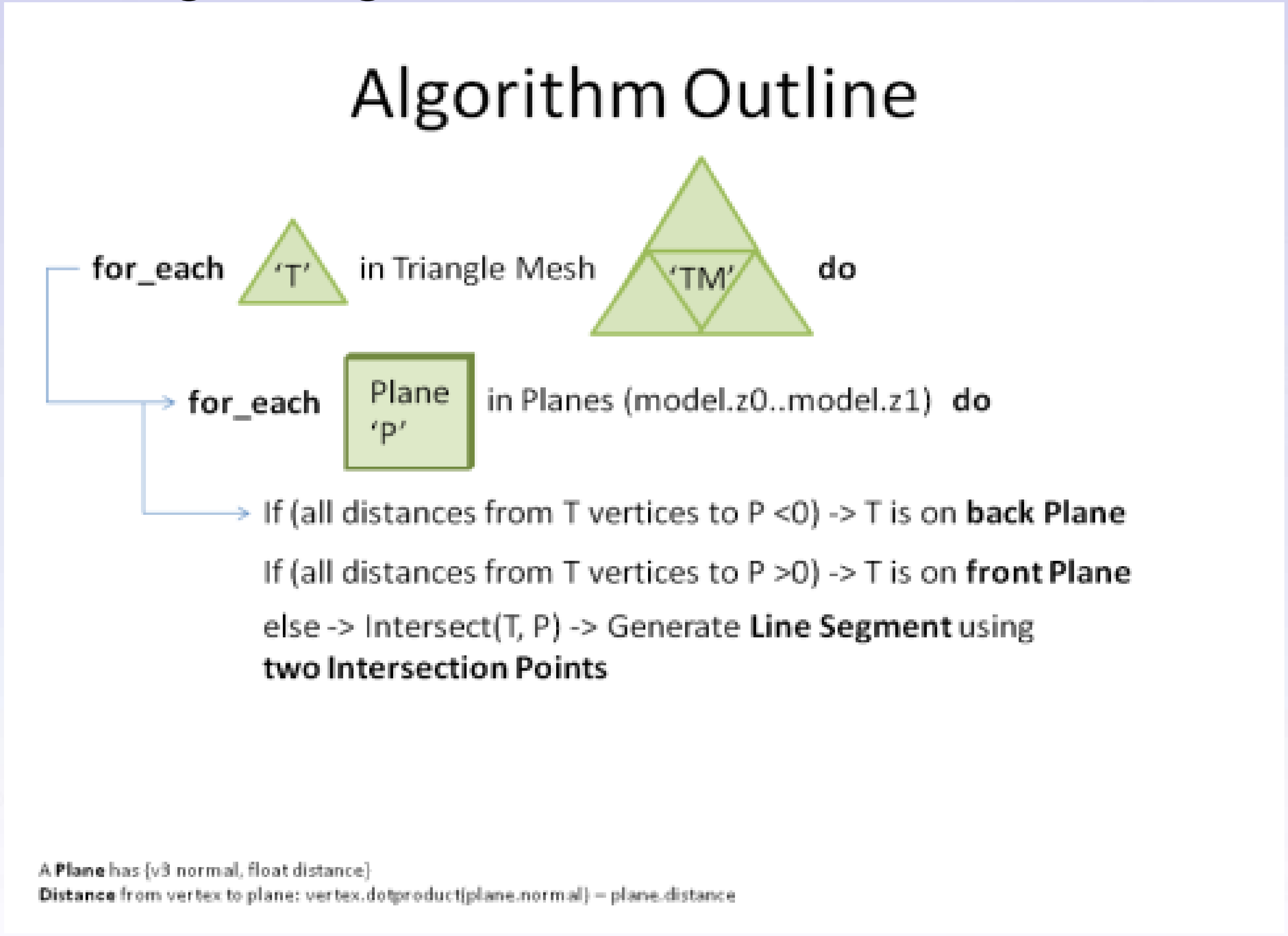

Triangle Slicing – 4 Cases

* There are more degenerate cases (The plane "falls" on one of the original vertices-->no quad generated)

Picture credit: Raveh Gonen

Software

Slicing – algorithm



Algorithm Outline

```
for_each 'T' in Triangle Mesh 'TM' do
    for_each Plane 'P' in Planes (model.z0..model.z1) do
        If (all distances from T vertices to P <0) -> T is on **back Plane**
        If (all distances from T vertices to P >0) -> T is on **front Plane**
        else -> Intersect(T, P) -> Generate **Line Segment** using **two Intersection Points**
```

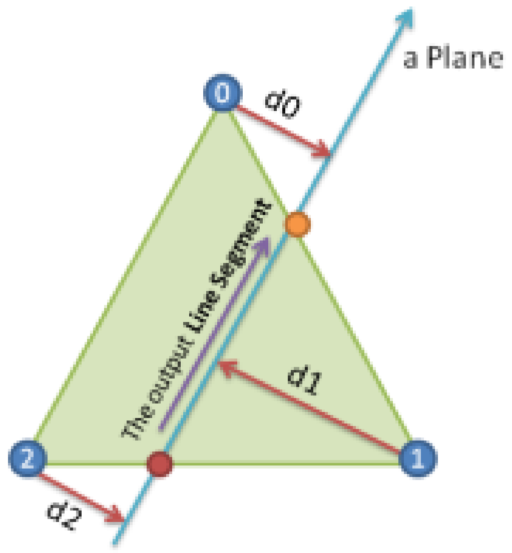
A **Plane** has {v3 normal, float distance}
Distance from vertex to plane: vertex.dotproduct(plane.normal) – plane.distance

Picture credit: Raveh Gonen

Software

Slicing – algorithm

Finding Triangle Plane Intersection



If (d0*d1<0)

s10 = d1/(d1-d0)   s21 = d2/(d2-d1)

● Intersection points = LinearInterp(1, 0, s10)

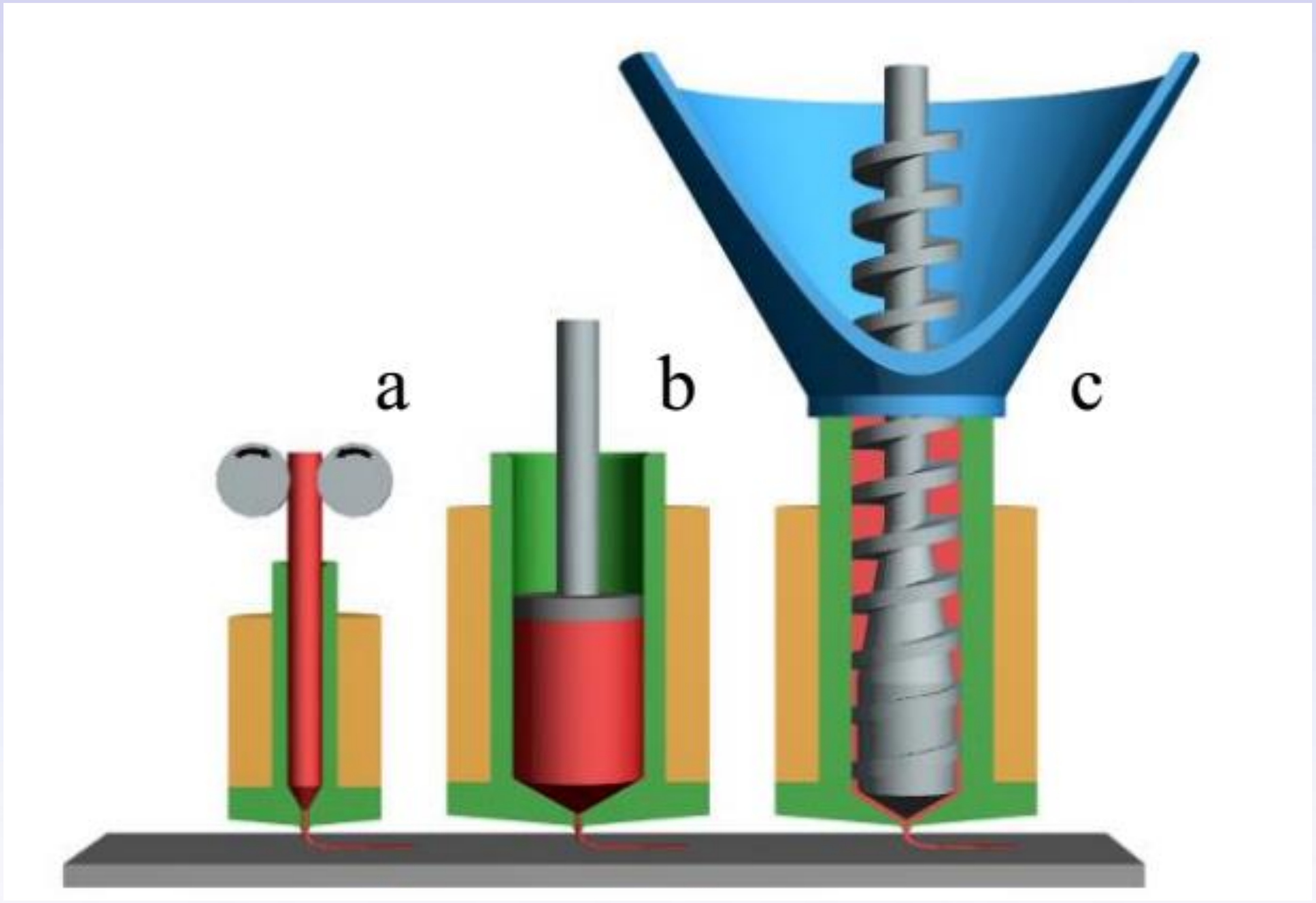
● Intersection points = LinearInterp(2, 1, s21)

A **Plane** has {v3 normal, float distance}
Distance from vertex to plane: vertex.dotproduct(plane.normal) – plane.distance

Picture credit: Raveh Gonen

Printhead

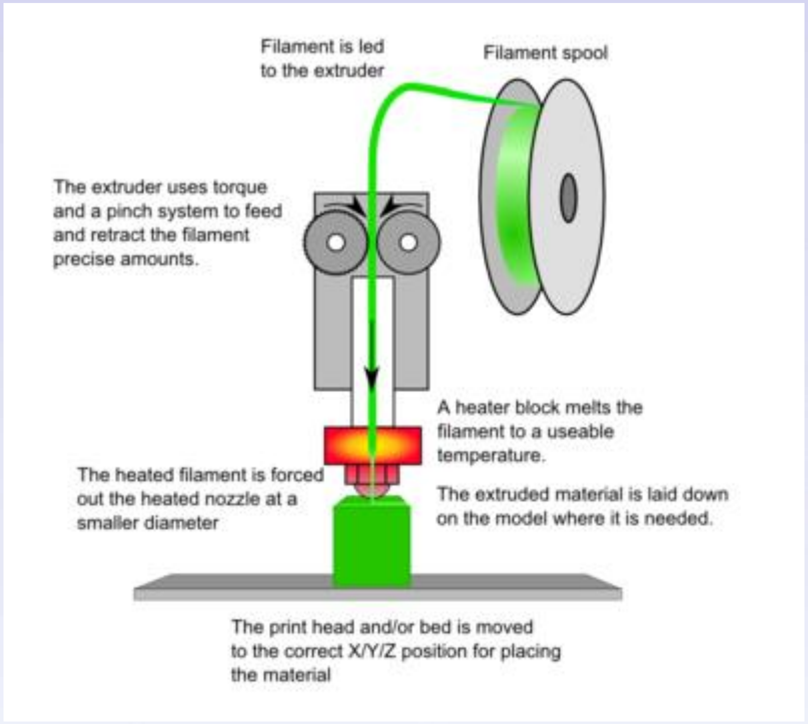
Extrusion



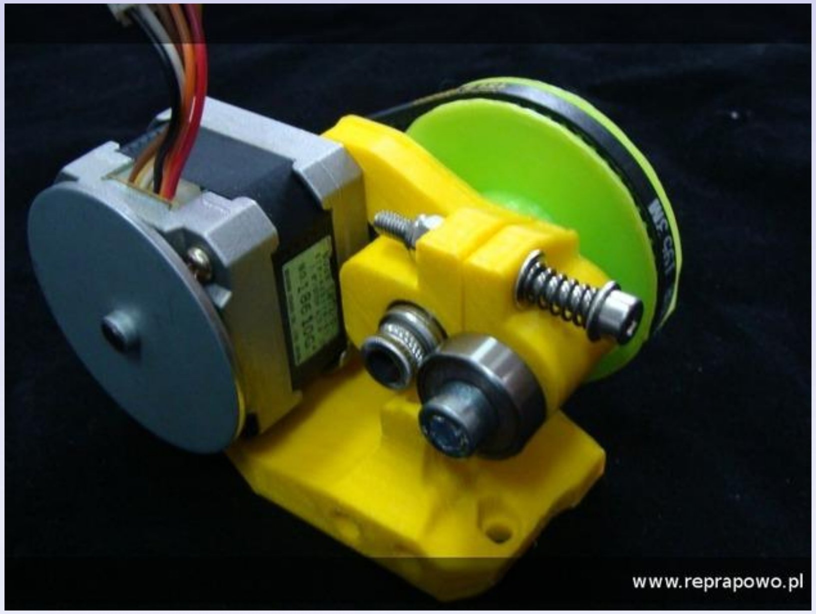
(a) Filament based extrusion; (b) Syringe based extrusion; (c) screw based extrusion

Printhead

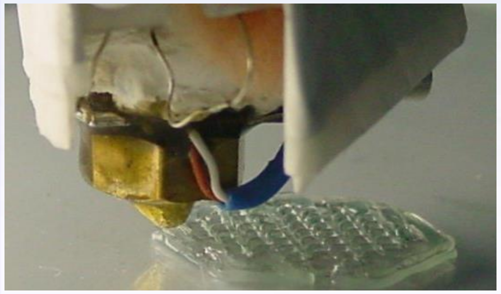
Extrusion



Filament is led to the extruder
Filament spool
The extruder uses torque and a pinch system to feed and retract the filament precise amounts.
A heater block melts the filament to a useable temperature.
The heated filament is forced out the heated nozzle at a smaller diameter
The extruded material is laid down on the model where it is needed.
The print head and/or bed is moved to the correct X/Y/Z position for placing the material

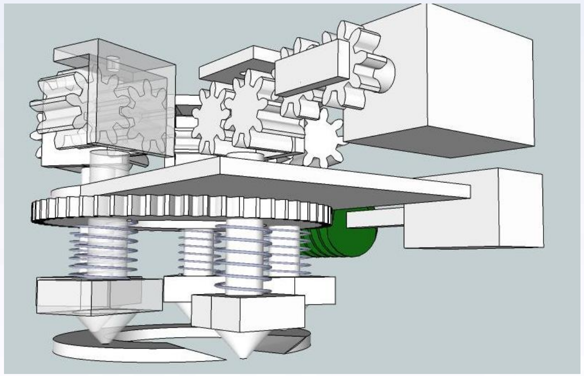


A simple design of cold end



A design of hot end

A compact design



A multi-head design

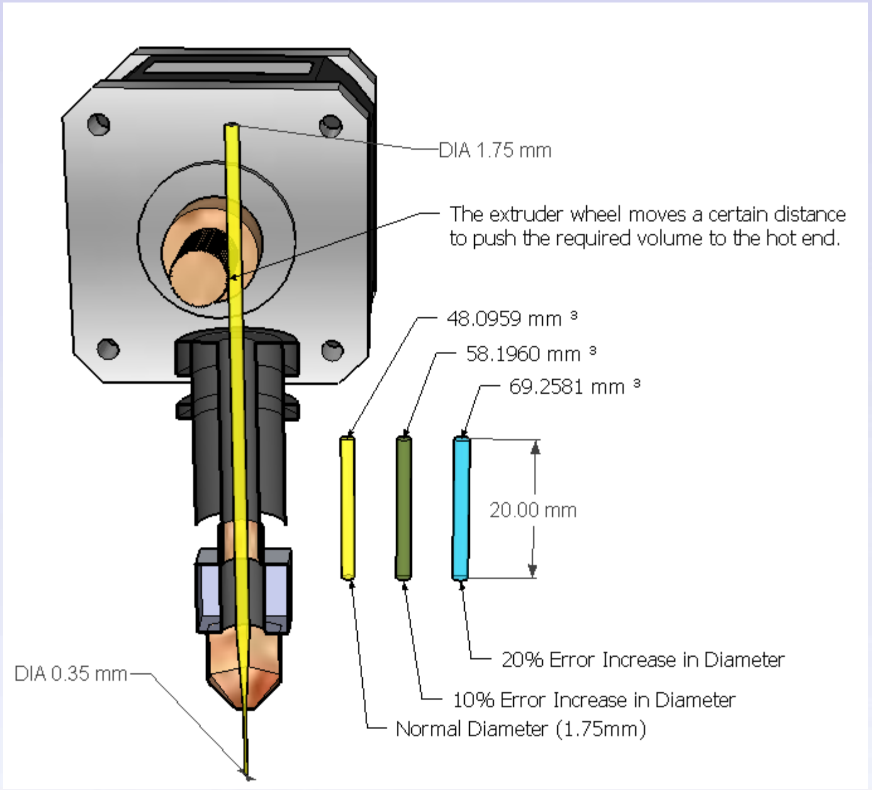
Pictures from google image

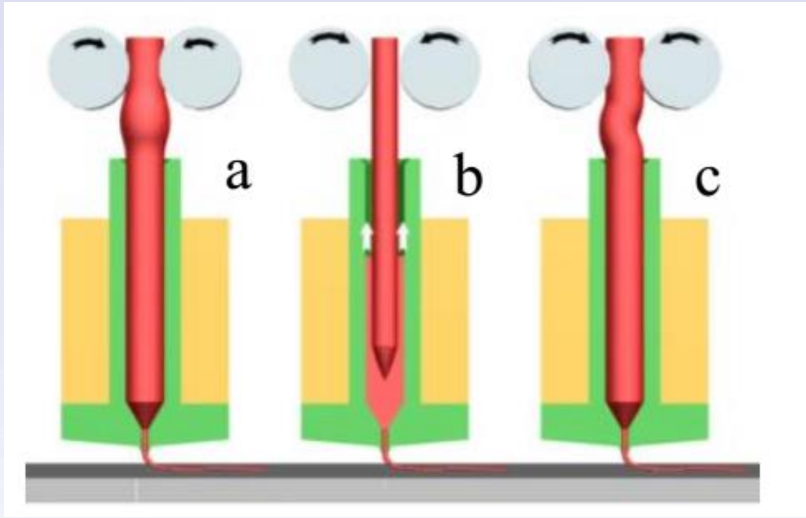
UNIVERSITY OF ARKANSAS

Printhead

AM³ Lab
Advanced Manufacturing | Modeling | Materials

FDM


DIA 1.75 mm
The extruder wheel moves a certain distance to push the required volume to the hot end.
48.0959 mm ³
58.1960 mm ³
69.2581 mm ³
20.00 mm
20% Error Increase in Diameter
10% Error Increase in Diameter
Normal Diameter (1.75mm)
DIA 0.35 mm



Problems caused by the filament: a), b) improper diameter filament, c) buckling

Control

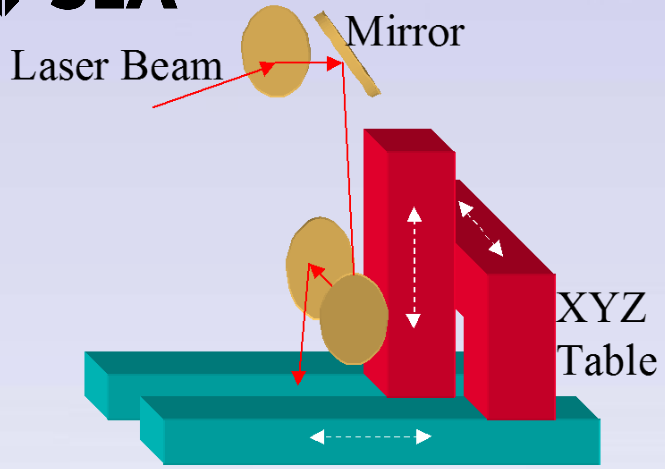
- **Extrusion temperature**
- **Extrusion speed**

UNIVERSITY OF ARKANSAS

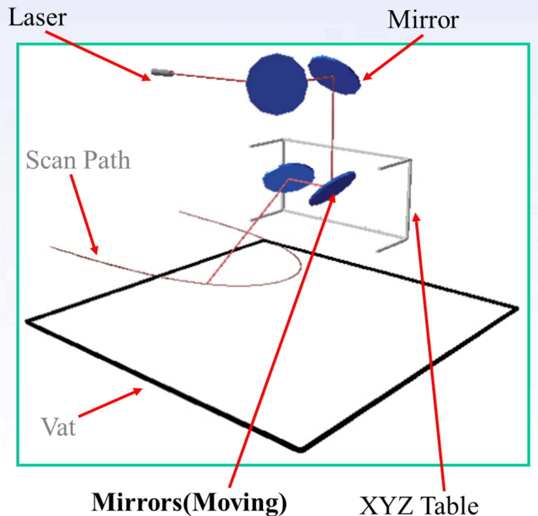
Printhead

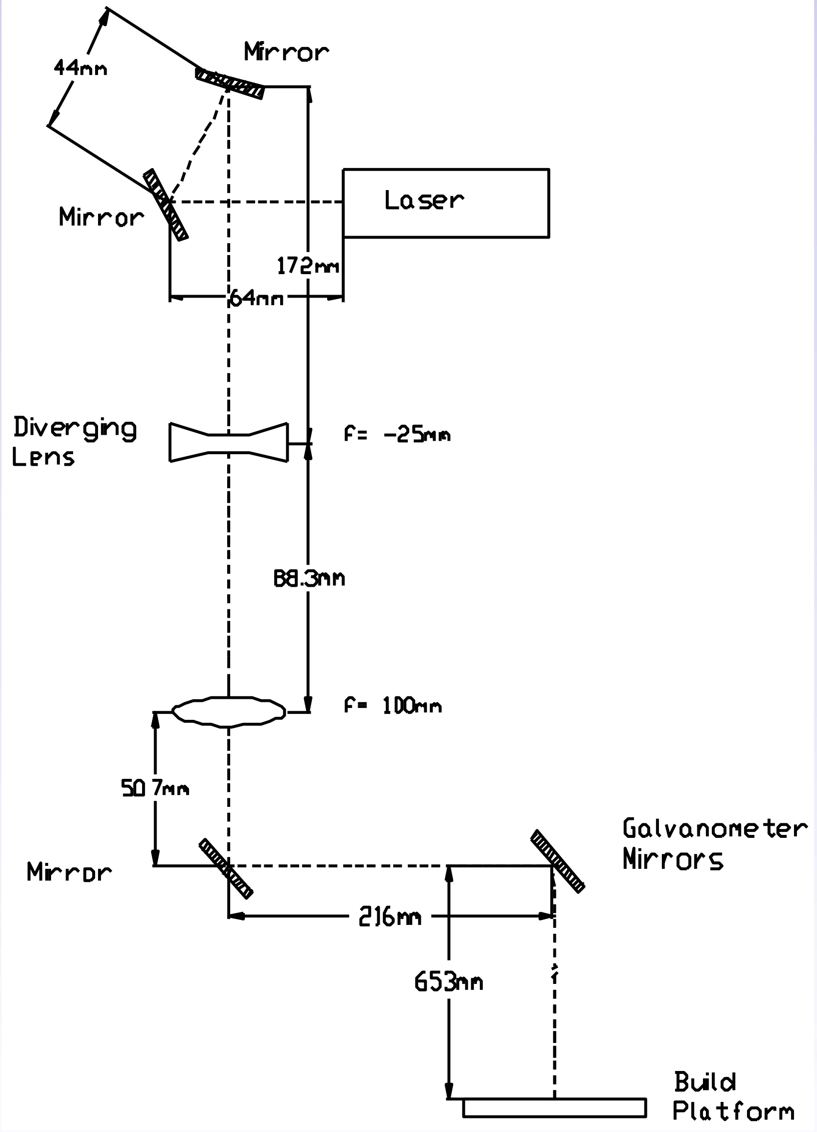
AM³ Lab
Advanced Manufacturing | Modeling | Materials

- SLA



4 Mirrors
- 2 stationary
- 2 mobile on XYZ table
- Total 7 DOF





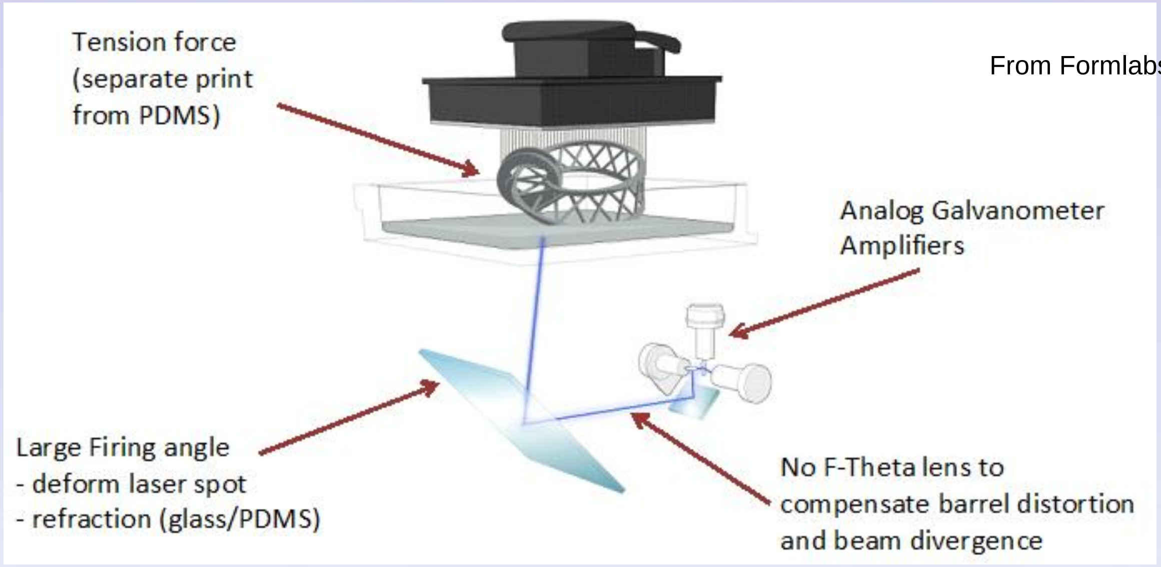
Courtesy: Dr. David Rosen

UNIVERSITY OF ARKANSAS

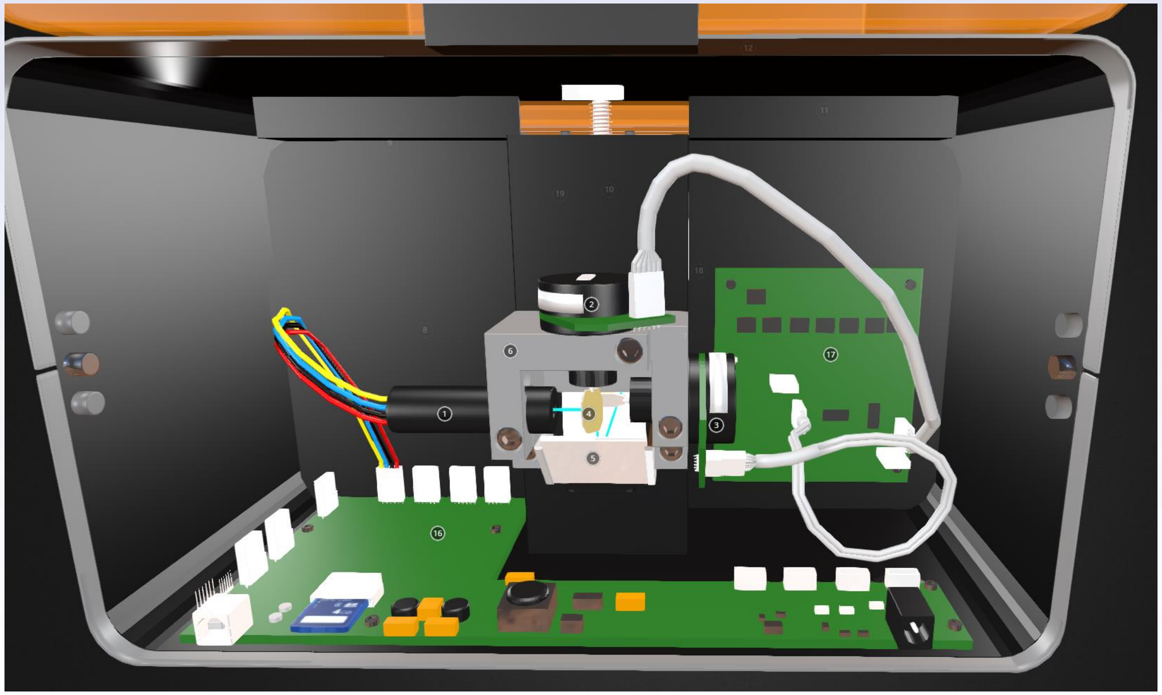
AM³ Lab
Advanced Manufacturing | Modeling | Materials

Printhead

- **SLA**

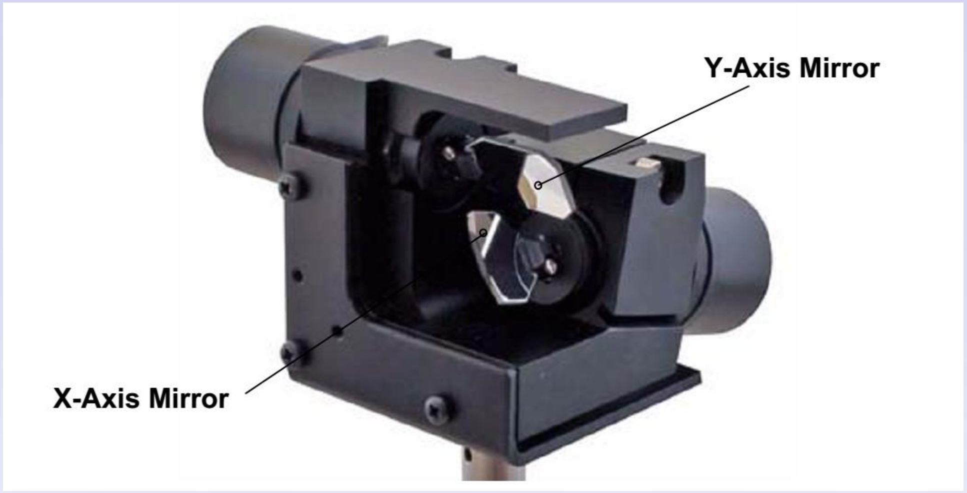

Tension force (separate print from PDMS)
Analog Galvanometer Amplifiers
Large Firing angle
- deform laser spot
- refraction (glass/PDMS)
No F-Theta lens to compensate barrel distortion and beam divergence

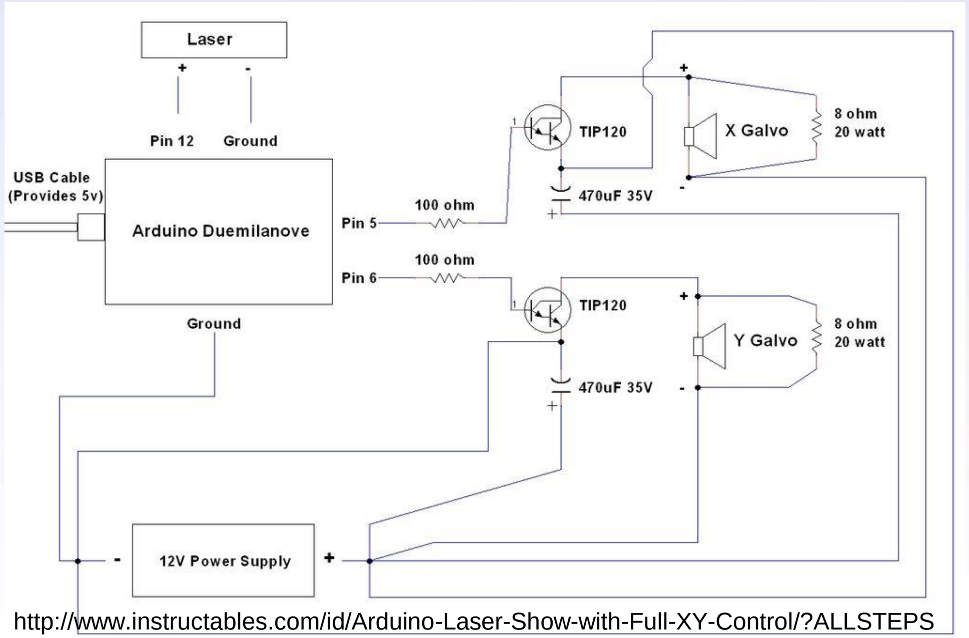
From Formlabs



Printhead

- SLA – control galvo mirror





http://www.instructables.com/id/Arduino-Laser-Show-with-Full-XY-Control/?ALLSTEPS

UNIVERSITY OF ARKANSAS

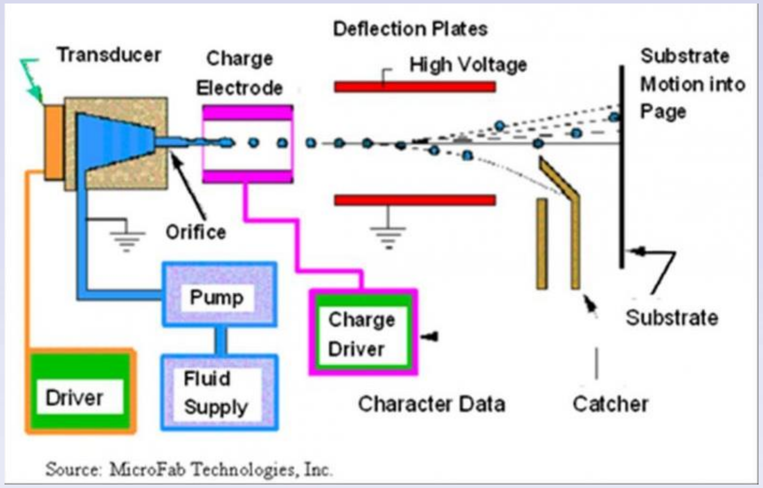
AM³ Lab
Advanced Manufacturing | Modeling | Materials

Printhead

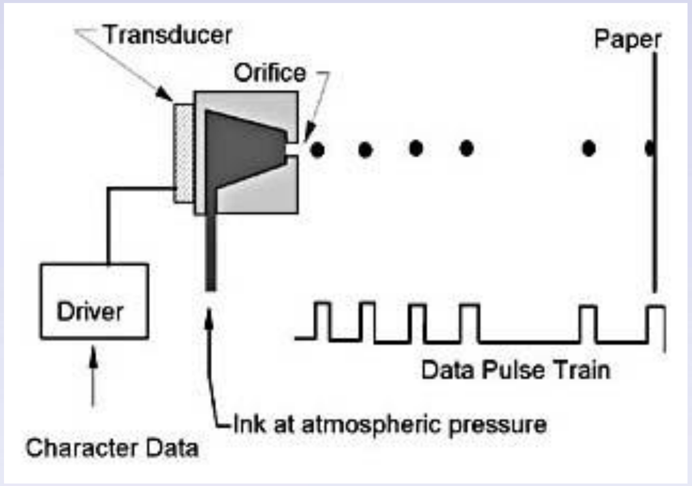
- Introduction
- What
- Why
- How
- Objectives
- Syllabus

Inkjet

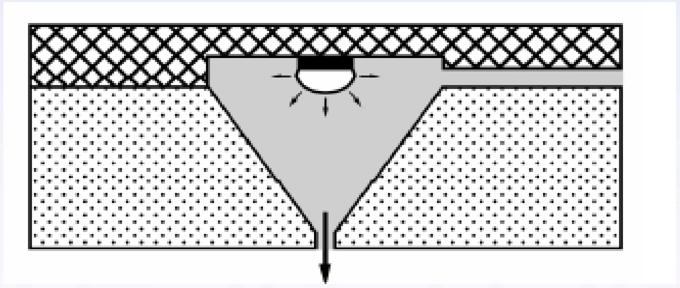
Continuous Inkjet (CIJ)

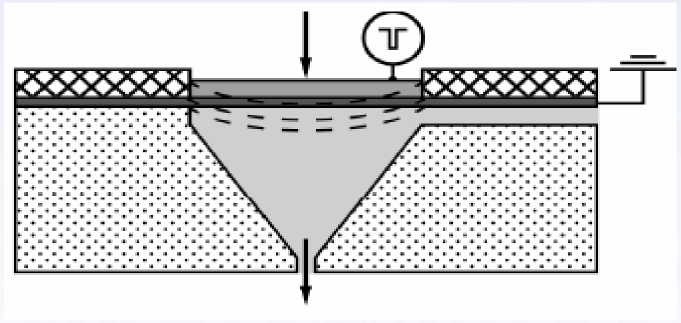


Drop-on-Demand (DoD) Inkjet



Common DoD Inkjets

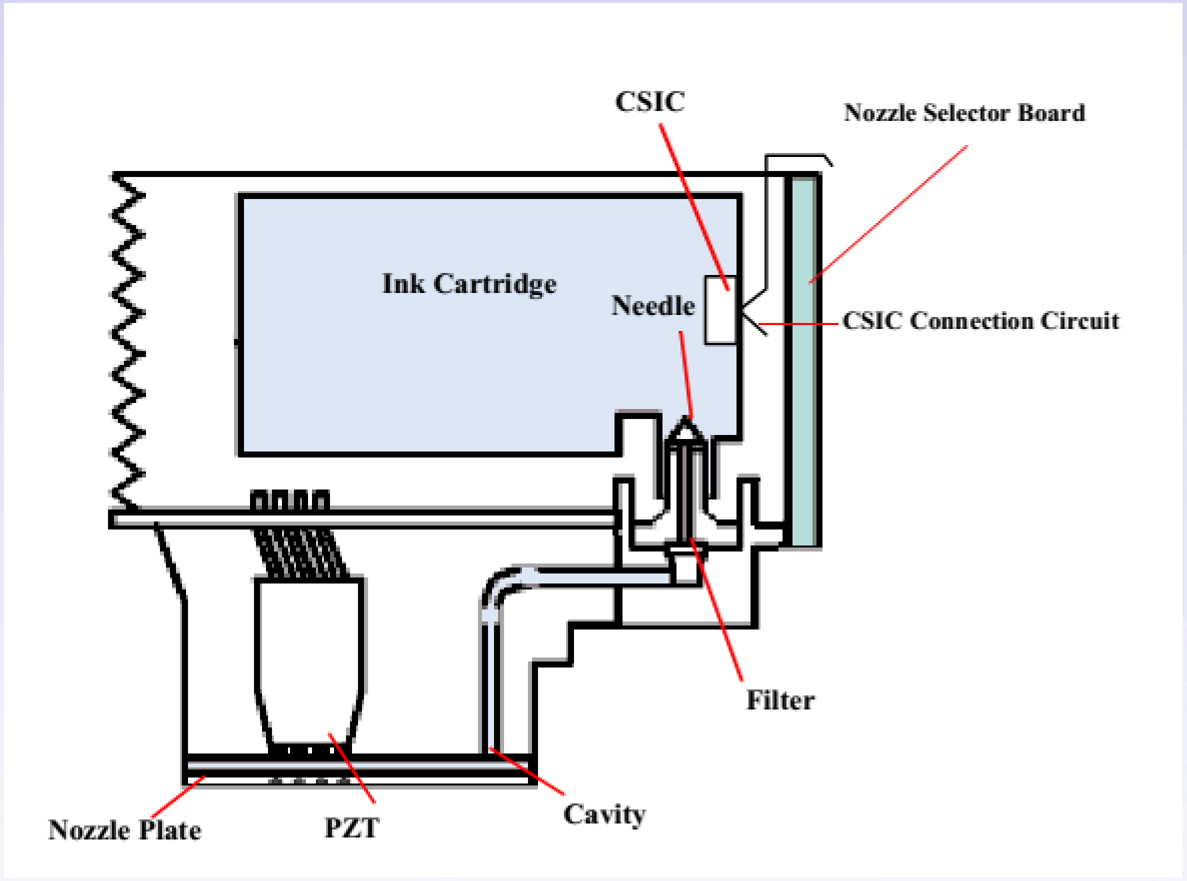

Thermo/Bubble jet: HP, Canon


Piezoelectric Inkjet: Epson

Viscosity limitation: up to ~40cP
Printing frequency: ~10 to ~100 kHz

Printhead

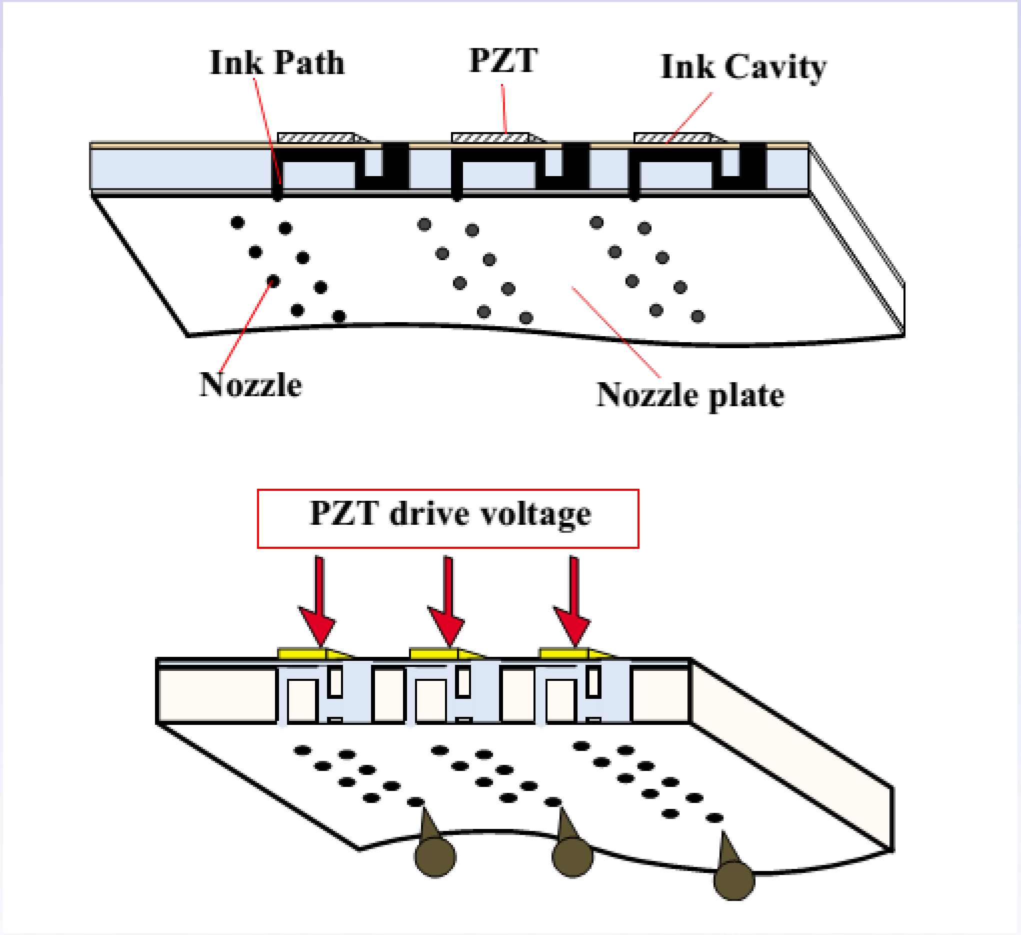
- **Inkjet**



Printhead sectional drawing (from Epson service manual)

Printhead

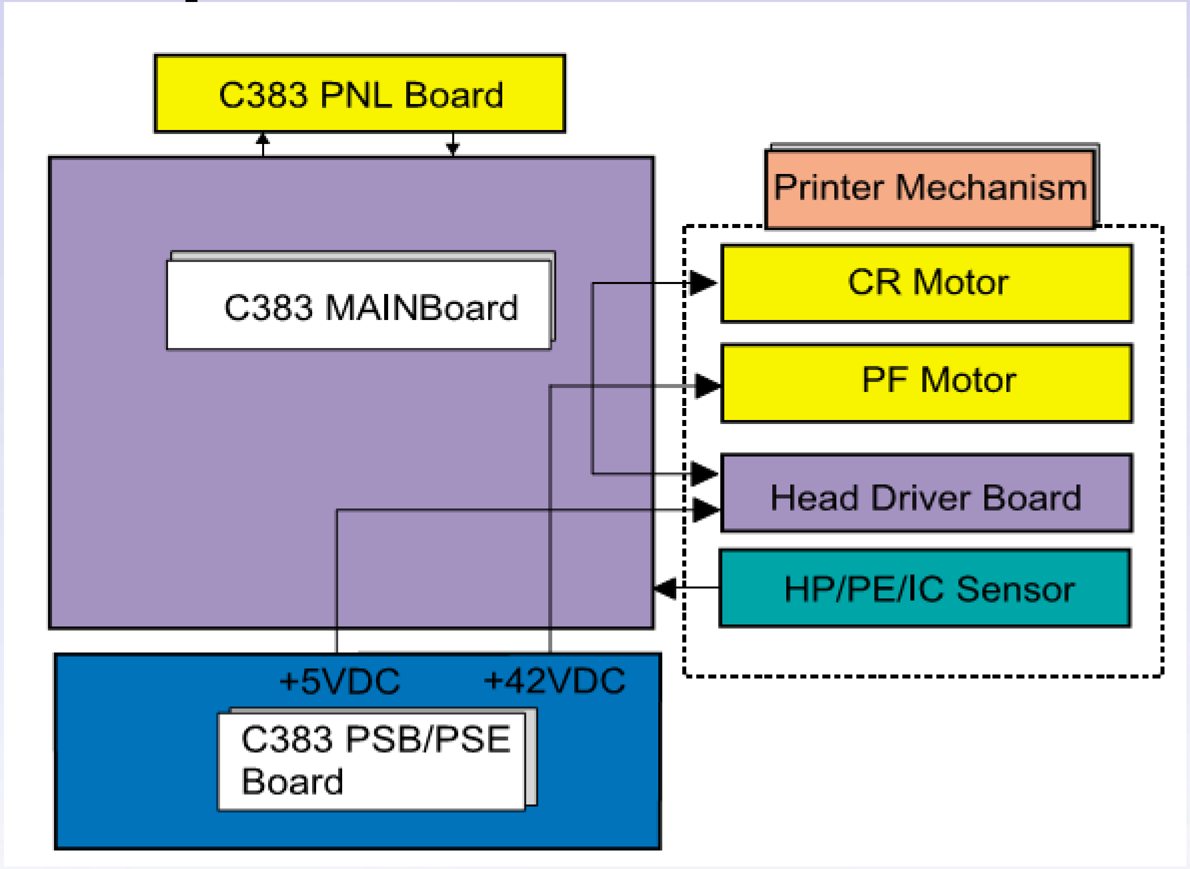
Inkjet



Printing process (from Epson service manual)

Printhead

Inkjet



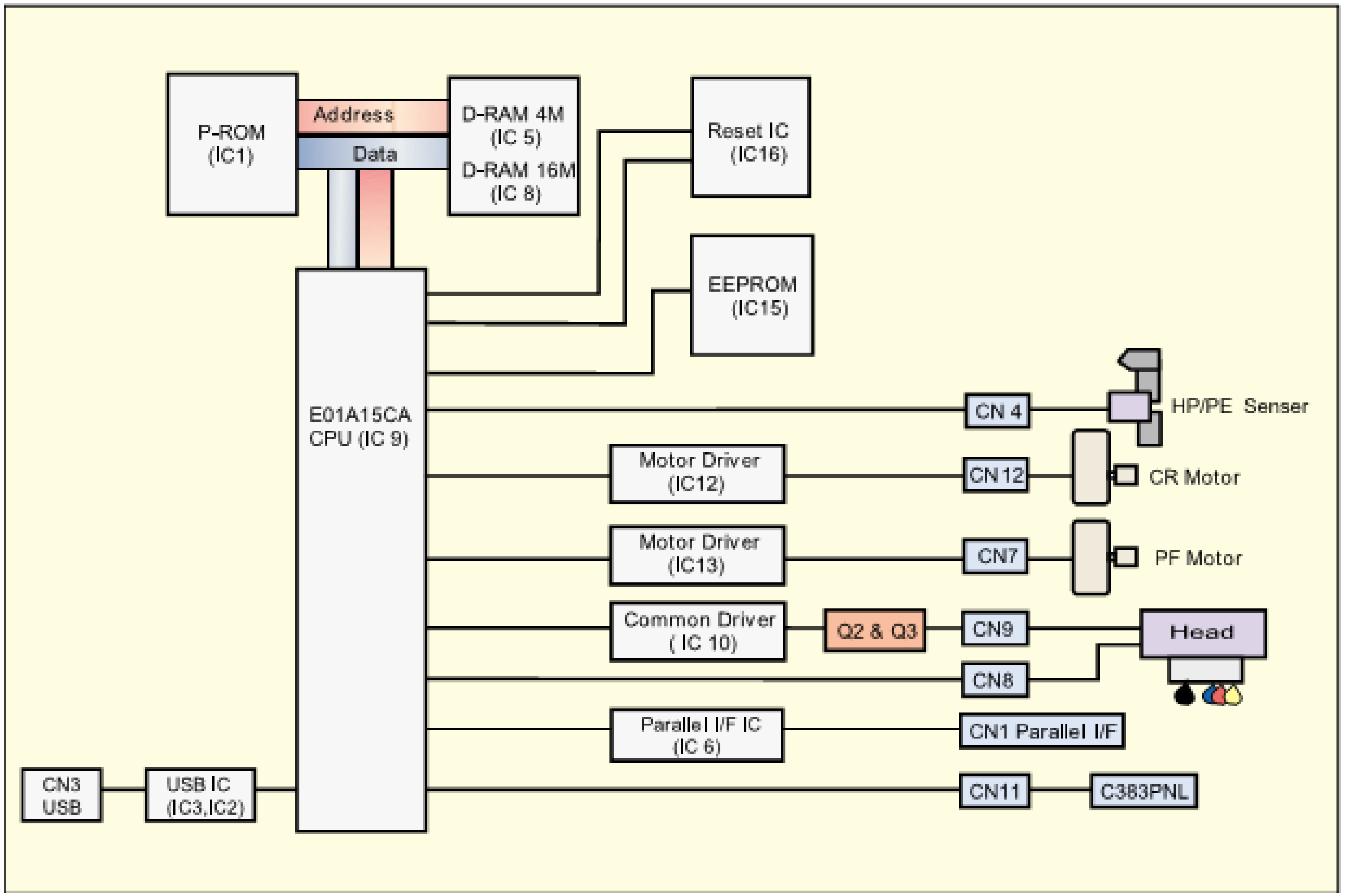
Carriage

Page feed

Electric circuit (from Epson service manual)

Printhead

Inkjet



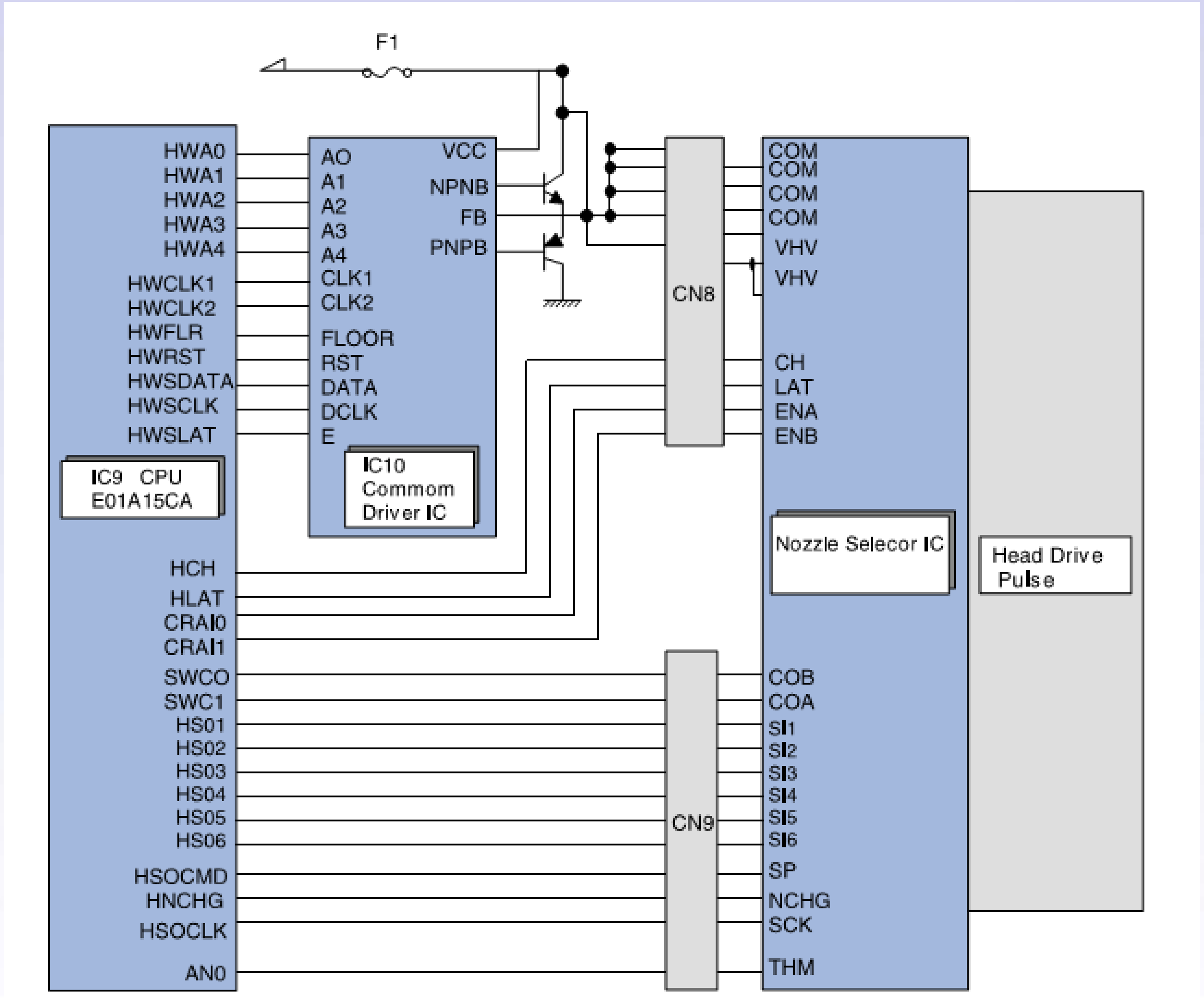
Block diagram C383 Mainboard (from Epson service manual)

UNIVERSITY OF ARKANSAS

Printhead

AM³ Lab
Advanced Manufacturing | Modeling | Materials

Inkjet



Printhead driver circuit (from Epson service manual)

THANK
YOU!