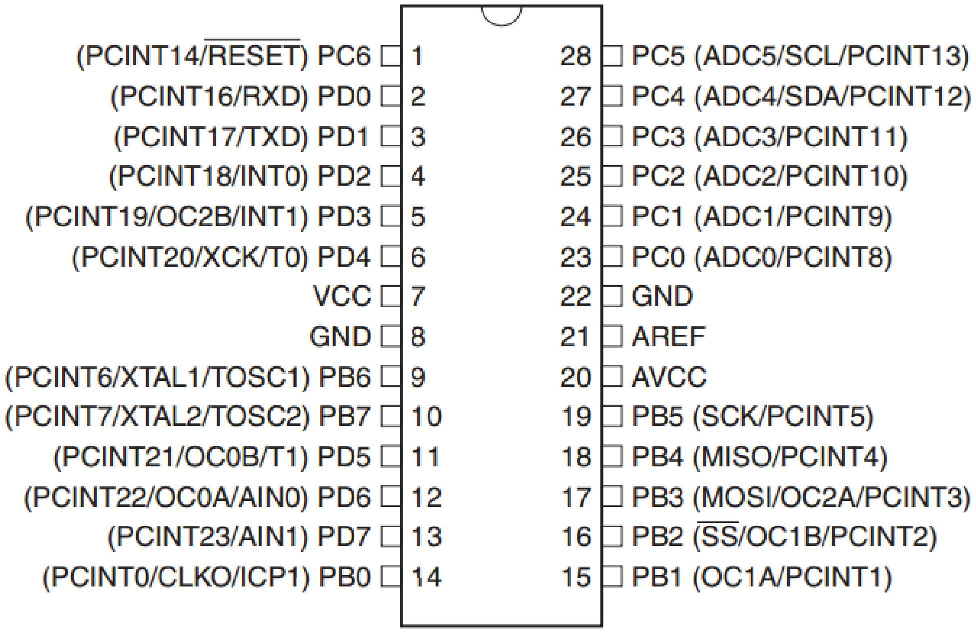
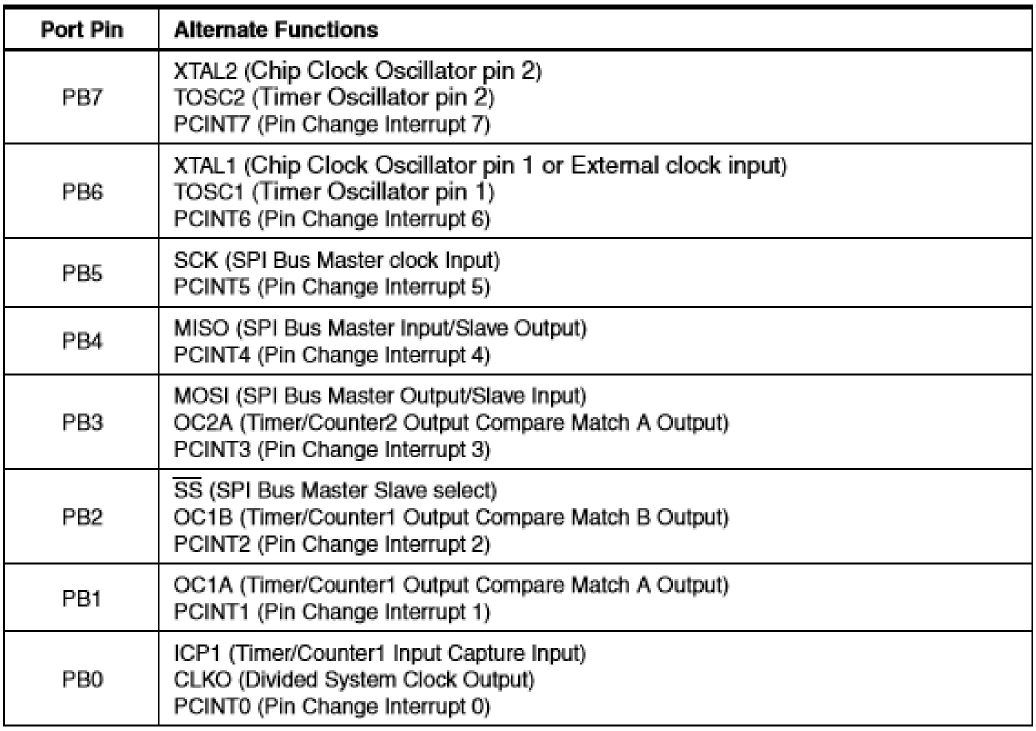
Konfigurasi PIN ATMega 328

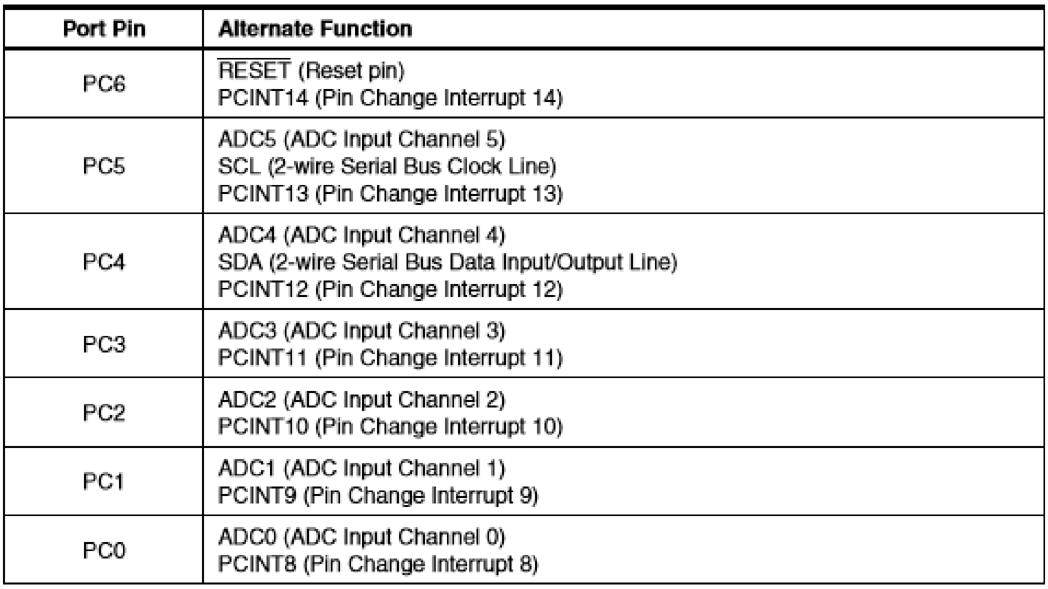


Konfigurasi Port B



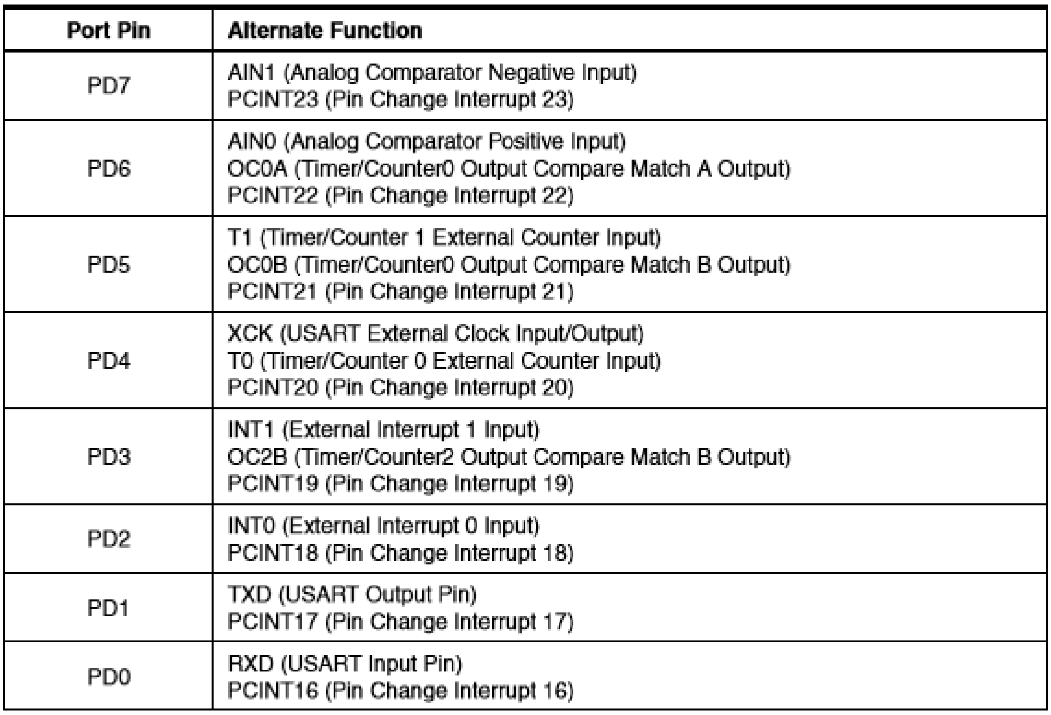
Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2) PCINT7 (Pin Change Interrupt 7)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1) PCINT6 (Pin Change Interrupt 6)
PB5	SCK (SPI Bus Master clock Input) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (SPI Bus Master Input/Slave Output) PCINT4 (Pin Change Interrupt 4)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2A (Timer/Counter2 Output Compare Match A Output) PCINT3 (Pin Change Interrupt 3)
PB2	$\overline{\text{SS}}$ (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output) PCINT2 (Pin Change Interrupt 2)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output) PCINT1 (Pin Change Interrupt 1)
PB0	ICP1 (Timer/Counter1 Input Capture Input) CLKO (Divided System Clock Output) PCINT0 (Pin Change Interrupt 0)

Konfigurasi Port C



Port Pin	Alternate Function
PC6	$\overline{\text{RESET}}$ (Reset pin) PCINT14 (Pin Change Interrupt 14)
PC5	ADC5 (ADC Input Channel 5) SCL (2-wire Serial Bus Clock Line) PCINT13 (Pin Change Interrupt 13)
PC4	ADC4 (ADC Input Channel 4) SDA (2-wire Serial Bus Data Input/Output Line) PCINT12 (Pin Change Interrupt 12)
PC3	ADC3 (ADC Input Channel 3) PCINT11 (Pin Change Interrupt 11)
PC2	ADC2 (ADC Input Channel 2) PCINT10 (Pin Change Interrupt 10)
PC1	ADC1 (ADC Input Channel 1) PCINT9 (Pin Change Interrupt 9)
PC0	ADC0 (ADC Input Channel 0) PCINT8 (Pin Change Interrupt 8)

Konfigurasi Port D



Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input) PCINT23 (Pin Change Interrupt 23)
PD6	AIN0 (Analog Comparator Positive Input) OC0A (Timer/Counter0 Output Compare Match A Output) PCINT22 (Pin Change Interrupt 22)
PD5	T1 (Timer/Counter 1 External Counter Input) OC0B (Timer/Counter0 Output Compare Match B Output) PCINT21 (Pin Change Interrupt 21)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input) PCINT20 (Pin Change Interrupt 20)
PD3	INT1 (External Interrupt 1 Input) OC2B (Timer/Counter2 Output Compare Match B Output) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	TXD (USART Output Pin) PCINT17 (Pin Change Interrupt 17)
PD0	RXD (USART Input Pin) PCINT16 (Pin Change Interrupt 16)

```
#include <Servo.h>
#include <AFMotor.h>
#define LINE_BUFFER_LENGTH 512

char STEP = MICROSTEP ;
// Servo posisi untuk naik dan turun
const int penZUp = 145;
const int penZDown = 83;
// Servo PWM menggunakan pin 10
const int penServoPin = 10 ;
// Stepper persteps
const int stepsPerRevolution = 48;
// membuat object servo untuk mengendalikannya
Servo penServo;
// Inisialisasi steppers untuk sumbu X dan Y dengan menggunakan pin Arduino
ini untuk menghubungkan ke H-L293D
AF_Stepper myStepperY(stepsPerRevolution, 1);
AF_Stepper myStepperX(stepsPerRevolution, 2);

/* Struktur, variabel global */
struct point {
  float x;
  float y;
  float z;
};
// Posisi kepala spidol saat ini
struct point actuatorPos;
//  Pengaturan gambar
float StepInc = 1;
int StepDelay = 0;
int LineDelay = 0;
int penDelay = 50;
// Langkah motor melaju 1 milimeter.
// Gunakan sketsa uji untuk menempelkan 100 langkah. Ukur panjang garis.
// Hitung langkah per mm. Masuk disinifloat StepsPerMillimeterX = 100.0;
float StepsPerMillimeterY = 100.0;
// Menggambar batas robot, dalam mm
float Xmin = -200;
float Xmax = 200;
float Ymin = -300;
float Ymax = 300;
float Zmin = 0;
float Zmax = 1;
float Xpos = Xmin;
float Ypos = Ymin;
float Zpos = Zmax;

/ / Set ke true untuk mendapatkan output debug.
boolean verbose = false;
// Perlu menafsirkan
// G1 untuk pindah
// G4 P300 (tunggu 150ms)
// M300 S30 (pena)
// M300 S50 (pen)

void setup() {
  //  Setup
  Serial.begin( 9600 );
  penServo.attach(penServoPin);
  penServo.write(penZUp);
  delay(100);
  // Turunkan jika perlu
  myStepperX.setSpeed(600);
  myStepperY.setSpeed(600);
   //   Set & pindah ke posisi default awal
   // TBD
   // Pemberitahuan !!!
  Serial.println("Mini Plotter CNC hidup dan berjalan!");
  Serial.print("X Jangkauan dari ");
  Serial.print(Xmin);
  Serial.print(" to ");
  Serial.print(Xmax);
  Serial.println(" mm.");
  Serial.print("Y Jangkauan dari ");
  Serial.print(Ymin);
  Serial.print(" to ");
  Serial.print(Ymax);
  Serial.println(" mm.");
}
void loop()
{
  delay(100);
  char line[ LINE_BUFFER_LENGTH ];
  char c;
  int lineIndex;
  bool lineIsComment, lineSemiColon;
  lineIndex = 0;
  lineSemiColon = false;
  lineIsComment = false;

  while (1) {
```

```
  // Penerimaan serial - Sebagian besar dari Grbl, menambahkan dukungan titik
koma
  while ( Serial.available() > 0 ) {
   c = Serial.read();
   if (( c == '\n') || (c == '\r') ) {           // Akhir baris mencapai
    if ( lineIndex > 0 ) {                       // Baris selesai Kemudian jalankan!
     line[ lineIndex ] = '\0';                   // Hentikan string
     if (verbose) {
      Serial.print( "Received : ");
      Serial.println( line );
     }
     processIncomingLine( line, lineIndex );
     lineIndex = 0;
    }
    else {
     // Baris kosong atau komentar Lewati blok
    }
    lineIsComment = false;
    lineSemiColon = false;
    Serial.println("ok");
   }
   else {
    if ( (lineIsComment) || (lineSemiColon) ) {   // Buang semua karakter
komentar
     if ( c == ')' )  lineIsComment = false;     // Akhir komentar Resume line
    }
    else {
     if ( c <= ' ' ) {                          // Buang garis dan kontrol karakter
     }
     else if ( c == '/' ) {                     // Block delete tidak didukung Abaikan
karakter.
```

```
          }
          else if ( c == '(' ) {                    // Aktifkan bendera komentar dan abaikan semua karakter sampai ')' atau EOL.
            lineIsComment = true;
          }
          else if ( c == ';' ) {
            lineSemiColon = true;
          }
          else if ( lineIndex >= LINE_BUFFER_LENGTH - 1 ) {
            Serial.println( "ERROR - lineBuffer overflow" );
            lineIsComment = false;
            lineSemiColon = false;
          }
          else if ( c >= 'a' && c <= 'z' ) {        // huruf besar huruf besar
            line[ lineIndex++ ] = c - 'a' + 'A';
          }
          else {
            line[ lineIndex++ ] = c;
          }
        }
      }
    }
  }
}

void processIncomingLine( char* line, int charNB ) {
  int currentIndex = 0;
  char buffer[ 64 ];                                // 64 sudah cukup untuk 1 parameter
  struct point newPos;
  newPos.x = 0.0;
  newPos.y = 0.0;
  // G1 untuk pindah
   // G4 P300 (tunggu 150ms)
   // G1 X60 Y30
   // G1 X30 Y50
   // M300 S30 (pena)
   // M300 S50 (pen)
   // buang apa saja dengan (
   // Buang perintah lain!
  while ( currentIndex < charNB ) {
    switch ( line[ currentIndex++ ] ) {              // Pilih perintah, jika ada
      case 'U':
        penUp();
        break;
      case 'D':
        penDown();
        break;
      case 'G':
        buffer[0] = line[ currentIndex++ ];          // bekerja dengan 2 digit perintah
        //      buffer[1] = line[ currentIndex++ ];
        //      buffer[2] = '\0';
        buffer[1] = '\0';

        switch ( atoi( buffer ) ) {                  // Pilih G command
          case 0:                                    // G00 & G01 - Gerakan atau gerakan cepat.
          case 1:
            // /!\ Dirty - Suppose that X is before Y
            char* indexX = strchr( line + currentIndex, 'X' ); // Get X/Y position in the string (if any)
            char* indexY = strchr( line + currentIndex, 'Y' );
            if ( indexY <= 0 ) {
              newPos.x = atof( indexX + 1);
              newPos.y = actuatorPos.y;
            }
            else if ( indexX <= 0 ) {
              newPos.y = atof( indexY + 1);
              newPos.x = actuatorPos.x;
            }
            else {
              newPos.y = atof( indexY + 1);
              indexY = '\0';
              newPos.x = atof( indexX + 1);
            }
            drawLine(newPos.x, newPos.y );
            //        Serial.println("ok");
            actuatorPos.x = newPos.x;
            actuatorPos.y = newPos.y;
            break;
        }
        break;
      case 'M':
        buffer[0] = line[ currentIndex++ ];        // /!\ Dirty - Only works with 3 digit
commands
        buffer[1] = line[ currentIndex++ ];
        buffer[2] = line[ currentIndex++ ];
        buffer[3] = '\0';
        switch ( atoi( buffer ) ) {
          case 300:
            {
              char* indexS = strchr( line + currentIndex, 'S' );
              float Spos = atof( indexS + 1);
              //        Serial.println("ok");
              if (Spos == 30) {
                penDown();
              }
              if (Spos == 50) {
                penUp();
              }
              break;
            }
          case 114:                                // M114 - Repport position
            Serial.print( "Absolute position : X = " );
            Serial.print( actuatorPos.x );
            Serial.print( "  -  Y = " );
            Serial.println( actuatorPos.y );
            break;
          default:
            Serial.print( "Command not recognized : M");
            Serial.println( buffer );
        }
    }
  }

}

void drawLine(float x1, float y1) {
  if (verbose)
  {
    Serial.print("fx1, fy1: ");
    Serial.print(x1);
    Serial.print(",");
    Serial.print(y1);
    Serial.println("");
  }
```

```
//  Bawa instruksi dalam batasan
if (x1 >= Xmax) {
  x1 = Xmax;
}
if (x1 <= Xmin) {
  x1 = Xmin;
}
if (y1 >= Ymax) {
  y1 = Ymax;
}
if (y1 <= Ymin) {
  y1 = Ymin;
}
if (verbose)
{
  Serial.print("Xpos, Ypos: ");
  Serial.print(Xpos);
  Serial.print(",");
  Serial.print(Ypos);
  Serial.println("");
}
if (verbose)
{
  Serial.print("x1, y1: ");
  Serial.print(x1);
  Serial.print(",");
  Serial.print(y1);
  Serial.println("");
}
```

```
//  Mengkonversi koordinat ke langkah-langkah
x1 = (int)(x1 * StepsPerMillimeterX);
y1 = (int)(y1 * StepsPerMillimeterY);
float x0 = Xpos;
float y0 = Ypos;

//  Mencari tahu perubahan koordinatnya
long dx = abs(x1 - x0);
long dy = abs(y1 - y0);
int sx = x0 < x1 ? StepInc : -StepInc;
int sy = y0 < y1 ? StepInc : -StepInc;
long i;
long over = 0;

if (dx > dy) {
  for (i = 0; i < dx; ++i) {
    myStepperX.onestep(sx, STEP);
    over += dy;
    if (over >= dx) {
      over -= dx;
      myStepperY.onestep(sy, STEP);
    }
    delay(StepDelay);
  }
}
else {
  for (i = 0; i < dy; ++i) {
    myStepperY.onestep(sy, STEP);
    over += dx;
    if (over >= dy) {
      over -= dy;
```

```
        myStepperX.onestep(sx, STEP);
      }
      delay(StepDelay);
    }
  }
  if (verbose)
  {
    Serial.print("dx, dy:");
    Serial.print(dx);
    Serial.print(",");
    Serial.print(dy);
    Serial.println("");
  }

  if (verbose)
  {
    Serial.print("Going to (");
    Serial.print(x0);
    Serial.print(",");
    Serial.print(y0);
    Serial.println(")");
  }

  //  Keterlambatan sebelum baris berikutnya diajukan
  delay(LineDelay);
  //  Update the positions
  Xpos = x1;
  Ypos = y1;
}
//  Menaikan pena
void penUp() {
  penServo.write(penZUp);
  delay(penDelay);
  Zpos = Zmax;
  digitalWrite(15, LOW);
  digitalWrite(16, HIGH);
  if (verbose) {
    Serial.println("Pen up!");

  }
}
//  Menurunkan pena
void penDown() {
  penServo.write(penZDown);
  delay(penDelay);
  Zpos = Zmin;
  digitalWrite(15, HIGH);
  digitalWrite(16, LOW);
  if (verbose) {
    Serial.println("Pen down.");


  }
}
```

```
import java.awt.event.KeyEvent;
import javax.swing.JOptionPane;
import processing.serial.*;

Serial port = null;

// pilih dan ubah baris yang sesuai untuk sistem operasi Anda
// (tekan 'p' dalam program)
String portname = null;
//String portname = "COM6"; // Windows

boolean streaming = false;
float speed = 0.001;
String[] gcode;
int i = 0;

void openSerialPort()
{
  if (portname == null) return;
  if (port != null) port.stop();

  port = new Serial(this, portname, 9600);

  port.bufferUntil('\n');
}

void selectSerialPort()
{
  String result = (String) JOptionPane.showInputDialog(frame,
    " Pilih port serial yang sesuai dengan Arduino Anda. ",
     "Pilih port serial",
    JOptionPane.QUESTION_MESSAGE,
    null,
    Serial.list(),
    0);

  if (result != null) {
    portname = result;
    openSerialPort();
  }
}

void setup()
{
  size(500, 250);
  openSerialPort();
}

void draw()
{
  background(0);
  fill(255);
  int y = 24, dy = 12;
  text("INSTRUCTIONS", 12, y); y += dy;
  text("p: pilih port serial", 12, y); y += dy;
  text("1: atur kecepatan to 0.001 inches (1 mil) per jog", 12, y); y += dy;
  text("2: atur kecepatan to 0.010 inches (10 mil) per jog", 12, y); y += dy;
  text("3: atur kecepatan to 0.100 inches (100 mil) per jog", 12, y); y += dy;
  text("", 12, y); y += dy;
  text("arrow keys: jog in x-y plane", 12, y); y += dy;
  text("page up & page down: jog in z axis", 12, y); y += dy;
  text("$: Pengaturan display grbl", 12, y); y+= dy;
  text("h: Kembali ke awal", 12, y); y += dy;
  text("0: (Mengatur lokasi spidol / pena)", 12, y); y += dy;
  text("g: masukan file data gcode", 12, y); y += dy;
  text("x: stop (this is NOT immediate)", 12, y); y += dy;
  y = height - dy;
  text("kecepatan yang digunakan: " + speed + " inches per step", 12, y); y -= dy;
  text("port serial yang digunakan: " + portname, 12, y); y -= dy;
}

void keyPressed()
{
  if (key == '1') speed = 0.001;
  if (key == '2') speed = 0.01;
  if (key == '3') speed = 0.1;

  if (!streaming) {
    if (keyCode == LEFT) port.write("G91\nG20\nG00 X-" + speed + " Y0.000 Z0.000\n");
    if (keyCode == RIGHT) port.write("G91\nG20\nG00 X" + speed + " Y0.000 Z0.000\n");
    if (keyCode == UP) port.write("G91\nG20\nG00 X0.000 Y" + speed + " Z0.000\n");
    if (keyCode == DOWN) port.write("G91\nG20\nG00 X0.000 Y-" + speed + " Z0.000\n");
    if (keyCode == KeyEvent.VK_PAGE_UP) port.write("G91\nG20\nG00 X0.000 Y0.000 Z" + speed + "\n");
    if (keyCode == KeyEvent.VK_PAGE_DOWN) port.write("G91\nG20\nG00 X0.000 Y0.000 Z-" + speed + "\n");
    if (key == 'h') port.write("G90\nG20\nG00 X0.000 Y0.000 Z0.000\n");
    if (key == 'v') port.write("$0=75\n$1=74\n$2=75\n");
    //if (key == 'v') port.write("$0=100\n$1=74\n$2=75\n");
    if (key == 's') port.write("$3=10\n");
    if (key == 'e') port.write("$16=1\n");
    if (key == 'd') port.write("$16=0\n");
    if (key == '0') openSerialPort();
    if (key == 'p') selectSerialPort();
    if (key == '$') port.write("$$\n");
  }

  if (!streaming && key == 'g') {
    gcode = null; i = 0;
    File file = null;
    println("Loading file...");
    selectInput("Select a file to process:", "fileSelected", file);
  }

  if (key == 'x') streaming = false;
}

void fileSelected(File selection) {
  if (selection == null) {
    println("Window was closed or the user hit cancel.");
  } else {
    println("User selected " + selection.getAbsolutePath());
    gcode = loadStrings(selection.getAbsolutePath());
    if (gcode == null) return;
    streaming = true;
    stream();
  }
}

void stream()
```

```
{
  if (!streaming) return;

  while (true) {
    if (i == gcode.length) {
      streaming = false;
      return;
    }

    if (gcode[i].trim().length() == 0) i++;
    else break;
  }

  println(gcode[i]);
  port.write(gcode[i] + '\n');
  i++;
}

void serialEvent(Serial p)
{
  String s = p.readStringUntil('\n');
  println(s.trim());

  if (s.trim().startsWith("ok")) stream();
  if (s.trim().startsWith("error")) stream(); // XXX: really?
}
```