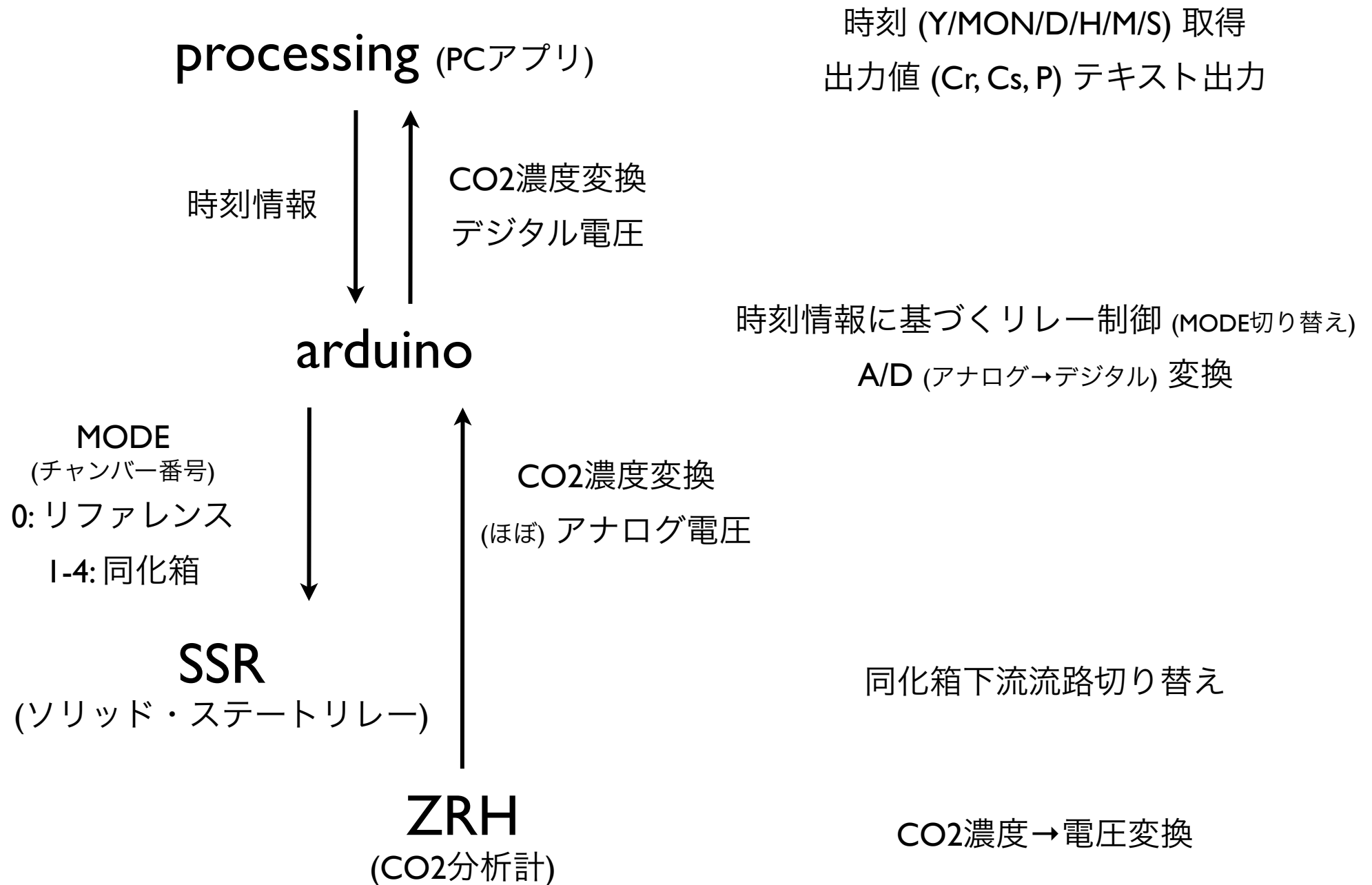


# モノおよび信号流れ

# 機能



# 測定の始め方

## 1. arduinoアップロード

1.1 CO2ガス用マスフロコントローラ開度 [mL/min] 設定

1.2 アップロード

## 2. 流入ガス調整

2.1 CO2ガスボンベ開

2.2 レイシーポンプ起動

2.3 マスフロコントローラ開度確認

2.4 加湿瓶確認

2.5 シリカゲル確認

2.6 ソーダライム確認

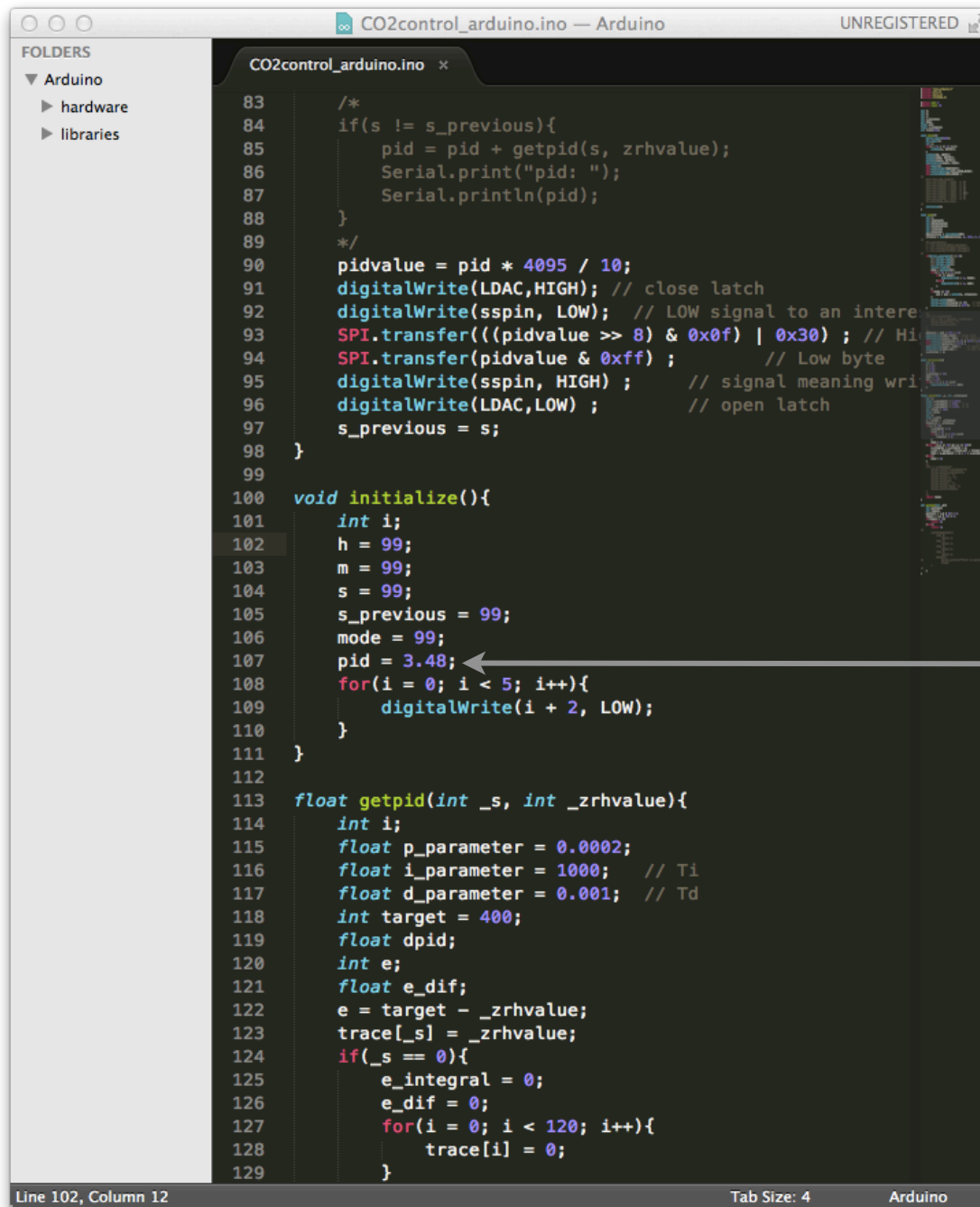
## 3. processingビルド

3.1 測定開始時刻設定

3.2 arduinoポート確認

3.3 ビルド

# I. arduinoアップロード (CO2control\_arduino.ino)



```
83  /*
84  if(s != s_previous){
85      pid = pid + getpid(s, zrhvalue);
86      Serial.print("pid: ");
87      Serial.println(pid);
88  }
89  */
90  pidvalue = pid * 4095 / 10;
91  digitalWrite(LDAC,HIGH); // close latch
92  digitalWrite(sspin, LOW); // LOW signal to an intere
93  SPI.transfer(((pidvalue >> 8) & 0x0f) | 0x30); // Hi
94  SPI.transfer(pidvalue & 0xff); // Low byte
95  digitalWrite(sspin, HIGH); // signal meaning wri
96  digitalWrite(LDAC,LOW); // open latch
97  s_previous = s;
98  }
99
100 void initialize(){
101     int i;
102     h = 99;
103     m = 99;
104     s = 99;
105     s_previous = 99;
106     mode = 99;
107     pid = 3.48;
108     for(i = 0; i < 5; i++){
109         digitalWrite(i + 2, LOW);
110     }
111 }
112
113 float getpid(int _s, int _zrhvalue){
114     int i;
115     float p_parameter = 0.0002;
116     float i_parameter = 1000; // Ti
117     float d_parameter = 0.001; // Td
118     int target = 400;
119     float dpid;
120     int e;
121     float e_dif;
122     e = target - _zrhvalue;
123     trace[_s] = _zrhvalue;
124     if(_s == 0){
125         e_integral = 0;
126         e_dif = 0;
127         for(i = 0; i < 120; i++){
128             trace[i] = 0;
129         }
130     }
```

I.1 CO2ガス用マスフロコントローラ  
開度 [mL/min] 設定

前回400 ppmだったときの数値  
を打ち込む

I.2 アップロード

option + command + U

### 3. processingビルド (CO2control\_processing.pde)

The screenshot displays the Processing IDE with the file `CO2control_processing.pde` open. The code is written in Java and includes imports for `processing.serial.*`, variable declarations for font, serial port, and various data structures, and a `void setup()` function. The `void setup()` function includes `size(490, 210);`, `frameRate(1);`, `font = loadFont("Calibri-24.vlw");`, `textFont(font);`, `textAlign(LEFT);`, `myPort = new Serial(this, "/dev/cu.usbmodem1421", 9600);`, `myPort.buffer(10);`, `initialize();`, and several `String` and `float` declarations and assignments. The status bar at the bottom indicates "Line 29, Column 26" and "Tab Size: 4 Processing".

On the right side, the Arduino IDE interface is visible, showing the "Tools" menu with the "Serial Port" option selected. The dropdown menu shows the selected port as `/dev/cu.usbmodem1421`. The status bar at the bottom of the Arduino IDE indicates "Line 102, Column 12" and "Tab Size: 4 Arduino".

3.1 測定開始時間設定  
5分刻みで

3.2 arduinoポート番号確認  
上図のように  
Arduinoタブ→Serial Port  
で✓以下の文字列を打ち込む

3.3 ビルド  
command + B

# 測定の終わり方

## 4. processing終了

4.1 processingウィンドウ左クリック→自動テキスト出力

4.2 テキストファイル確認

## 5. ガス片付け

5.1 CO<sub>2</sub>ガスボンベ閉

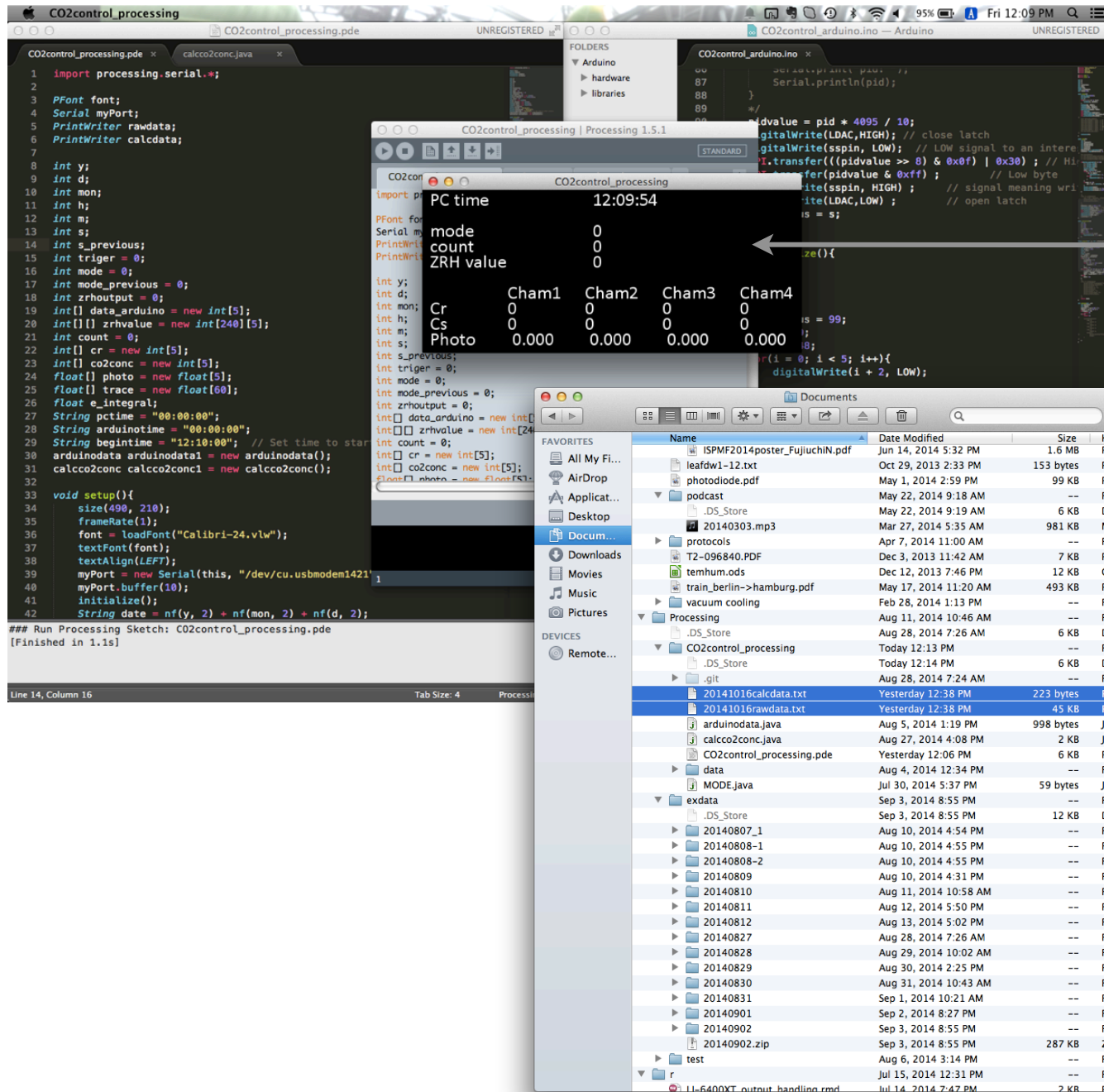
5.2 レイシーポンプ停止

## 6. arduino再アップロード

6.1 CO<sub>2</sub>ガス用マスフロコントローラ開度 [mL/min] 設定

6.2 アップロード

## 4. processing終了



4.1 ウィンドウを  
左クリック  
→自動でテキスト出力

4.2 テキストファイル確認

Finder  
Document  
Processing  
CO2control\_  
processing  
内に測定開始日の  
ファイル2つ

## 6. arduino再アップロード (CO2control\_arduino.ino)

```
83  /*
84  if(s != s_previous){
85      pid = pid + getpid(s, zrhvalue);
86      Serial.print("pid: ");
87      Serial.println(pid);
88  }
89  */
90  pidvalue = pid * 4095 / 10;
91  digitalWrite(LDAC,HIGH); // close latch
92  digitalWrite(sspin, LOW); // LOW signal to an intere
93  SPI.transfer(((pidvalue >> 8) & 0x0f) | 0x30); // Hi
94  SPI.transfer(pidvalue & 0xff); // Low byte
95  digitalWrite(sspin, HIGH); // signal meaning wri
96  digitalWrite(LDAC,LOW); // open latch
97  s_previous = s;
98  }
99
100 void initialize(){
101     int i;
102     h = 99;
103     m = 99;
104     s = 99;
105     s_previous = 99;
106     mode = 99;
107     pid = 3.48;
108     for(i = 0; i < 5; i++){
109         digitalWrite(i + 2, LOW);
110     }
111 }
112
113 float getpid(int _s, int _zrhvalue){
114     int i;
115     float p_parameter = 0.0002;
116     float i_parameter = 1000; // Ti
117     float d_parameter = 0.001; // Td
118     int target = 400;
119     float dpid;
120     int e;
121     float e_dif;
122     e = target - _zrhvalue;
123     trace[_s] = _zrhvalue;
124     if(_s == 0){
125         e_integral = 0;
126         e_dif = 0;
127         for(i = 0; i < 120; i++){
128             trace[i] = 0;
129         }
130     }
```

6.1 CO2ガス用マスフロコントローラ  
開度 [mL/min] 設定

前回400 ppmだったときの数値  
を打ち込む

6.2 アップロード

option + command + U