# 電子工作 temperature collector

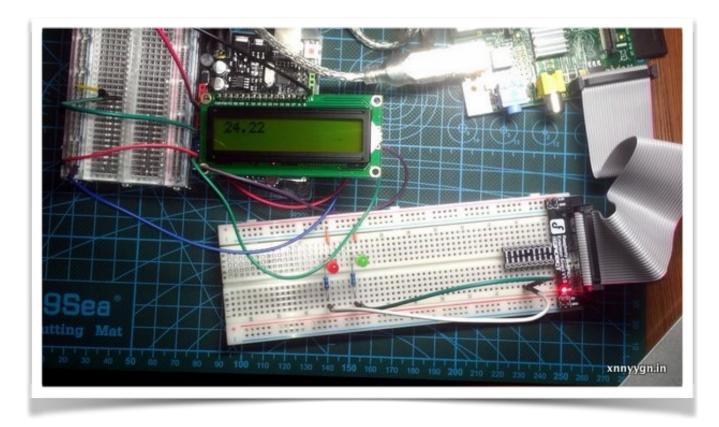
XnnYygn 2014-11-09

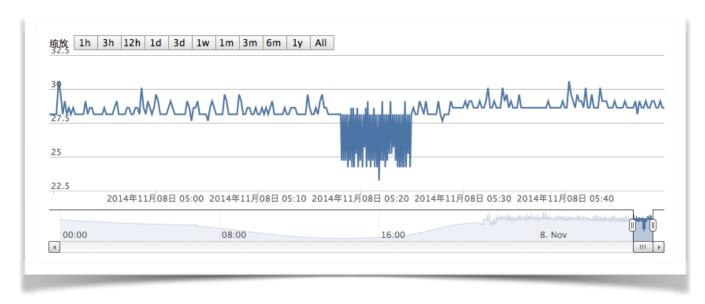
# 1. overview

# 1.1 keyword

arduino, raspberry pi, lm35, lcd, network, yeelink

# 1.2 presentation





#### 1.3 how it works

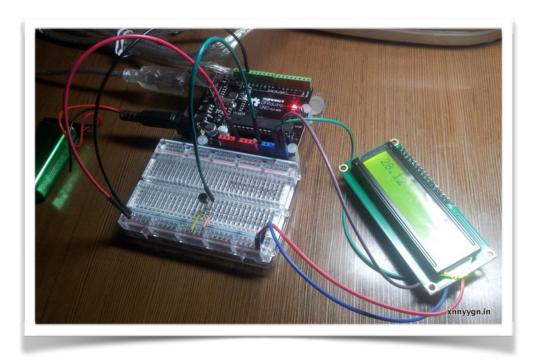
この作品は全部自動的です。

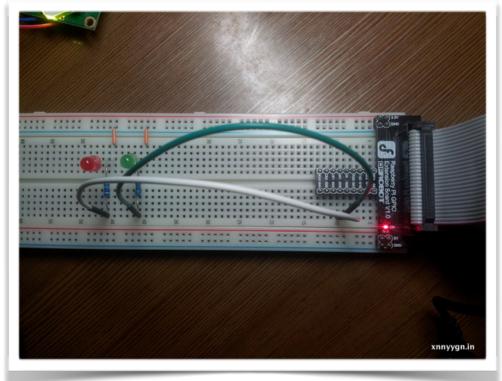
今の温度を収集してネットに転送します。そして、yeelinkというサイトでそのデータを分析します。

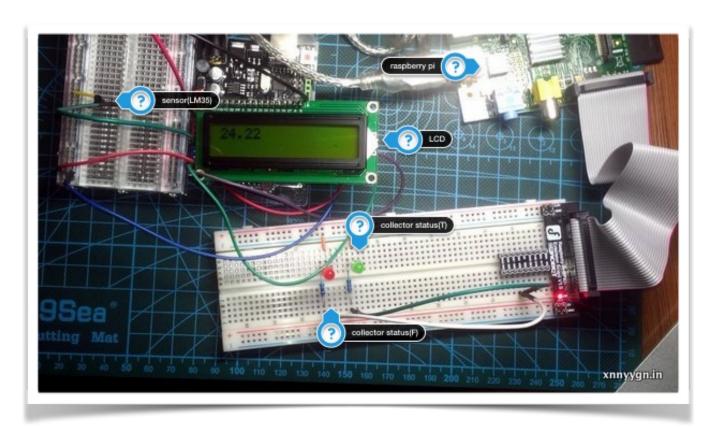
LCDは今の温度を表示します。

展示にある2つのLEDは転送する状態を示すためです。もし転送が成功する場合、緑のLEDがつきます。失敗なら赤いの方です。

# 1.4 pictures







```
iTerm Shell Edit View Profiles Toolbelt Window Help
                                                                                                                                                                                                                                                                                                                    >> ♥ 60 ▲ ♥ ¶ * ■ 🛜 🖷 99% 🖦 Fri Nov 7 21:14
                                                                                                                                                                                                                                                                        sensor data collector start
2014-11-07T21:09:40.900870 27.15
2014-11-07T21:09:53.043955 26.17
2014-11-07T21:10:05.185238 26.17
   14 def loop(s):
15 s.writelines(['G']) # send serial
16 c = s.readline().strip
17 if not c:
 2014-11-07T21:10:05.185238 26.17
2014-11-07T21:10:17.328232 26.17
2014-11-07T21:10:17.328232 26.17
2014-11-07T21:10:29.469188 26.17
2014-11-07T21:10:39.35464 26.17
2014-11-07T21:11:05.897552 26.17
2014-11-07T21:11:30.37744 26.17
2014-11-07T21:11:30.37744 26.17
2014-11-07T21:11:30.37744 26.17
2014-11-07T21:11:42.322007 26.17
2014-11-07T21:11:42.322007 26.17
2014-11-07T21:12:30.35154 26.17
2014-11-07T21:12:30.391534 26.17
2014-11-07T21:12:43.034718 26.17
2014-11-07T21:12:55.175807 26.17
2014-11-07T21:12:55.175807 26.17
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        7
                      if not c:
    return
now = datetime.datetime.now().isoformat()
print '[INFO]', now, c
if send_sensor_data(now, c):
    GPIO.output(PIN_LED_PASSED, GPIO.HIGH)
                                                                                                                                                                                                                                                     [INFO]
[INFO]
[INFO]
                                 GPIO.output(PIN_LED_FAILED, GPIO.HIGH)
                      sleep(1)

GPIO.output(PIN.LED_PASSED, GPIO.LOW)

GPIO.output(PIN.LED_FAILED, GPIO.LOW)

sleep(10) # sleep 10 seconds for yeelink limit(> 10s after previous request)
                       send_sensor_data(timestamp, value):
payload = '{"timestamp":"' + timestamp + '","value":' + value + '}'
headers = {'U-ApiKey': API_KEY}
                                                                                                                                                                                                                                                    [INFO] 2014-11-07721:12:55.175807 26.17

[INFO] 2014-11-07721:13:19.460165 26.17

[INFO] 2014-11-07721:13:31.604234 26.66

[INFO] 2014-11-07721:13:35.604234 26.66

[INFO] 2014-11-07721:13:55.888505 26.66

[INFO] 2014-11-07721:14:08.028660 26.66
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         S.
                      conn = httplib.HTTPConnection(HOST)
conn.request('POST', URI, payload, headers)
resp = conn.getresponse()
conn.close()
if resp.status == 200:
                                 print '[WARN] failed to send sensor data,', resp.reason return False
                 erature-collector.py" 57L, 1580C written
  one.
Jisconnecting from 192.168,2.107:955... done.
knnyygns-MacBook-Air:raspbery-pi xnnyygn$ ./upload temperature-collector.py
[raspberry-pi] Executing task 'upload'
[raspberry-pi] put: temperature-collector.py -> ./temperature-collector.py
Disconnecting from 192.168,2.107:955... done.
xnnyygns-MacBook-Air:raspbery-pi xnnyygn$
xnnyygn 0:goagent- 1:ssh*
```

# 2. design

もし電子工作の作品にネットの機能がつけば、とても面白いになるんでしょう。この作品は一番 簡単な温度を試しました。

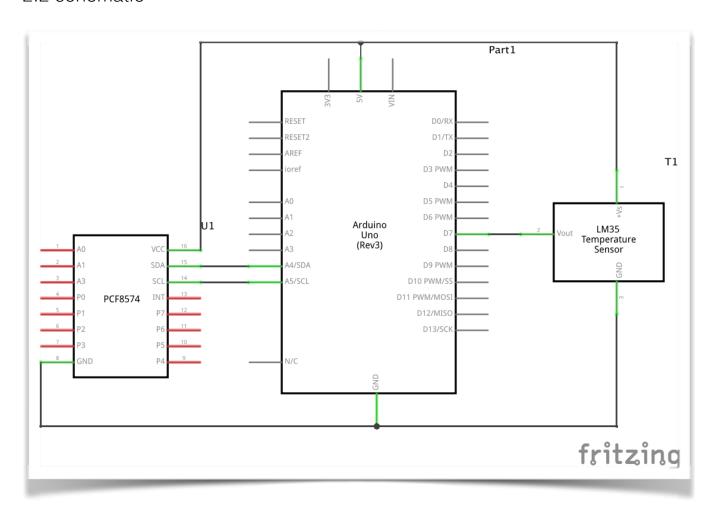
開発を選んだ時、やっぱりarduinoとraspberry piの組み合わせでいいかもしれない。raspberry pi は直接アナログの入力ができないので、敢えて使えば、ADC(analog digital converter)が必要で す。それと、ADC普通にはI2Cのインターフェイスなどがあります。LCDもI2Cなので、一緒に使えるADCを選んでなければなりません。

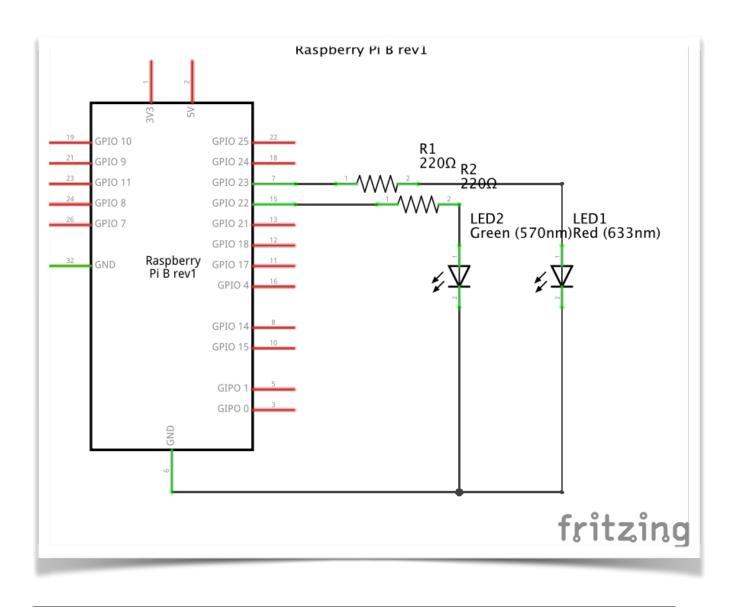
もう一方、アナログのデータを読めますが、ardunioはネットの機能がついていません。ネットのシールドは私が持っていないから、前回と同じ、2つのボードを利用してまらいました。データ収集はyeelinkというサイトに任せます。自分は時間と温度だけを転送すればいいです。

# 2.1 parts

name	description
raspberry pi x 1	main controller
arduino uno x 1	
lm35(temperature sensor) x 1	
lcd1602 with i2c	
led(red) x 1	
led(green) x 1	
resistance 560 x 2	

#### 2.2 schematic





# 2.3 pin data sheet

part	pin from	pin to(arduino/rapsberry-pi)
LCD 1602 with I2C	VCC	5V
	GND	GND
	SDA	A4
	SCL	A5
LM35(temperature sensor)	VCC	5V
	GND	GND
	OUT	D7
LED(red)	(+)	GPIO22
	(-)	GND
LED(green)	(+)	GPIO23
	(-)	GND

Page 5 of 8

```
2.4 protocol arduinoとraspberry piの通信は前回と同じUSBでserialコミニュケーションです。
raspberry pi -> arduino

G arduino -> raspberry pi (reply)

22.43(current temperature)
次はraspberry piとサーバーの通信です。
yeelinkのドキュメントによって、センサーとサーバーの通信はHTTPで行います。

POST http://api.yeelink.net/v1.0/device/<device_id>/sensor/<sensor_id>/datapoints

HEADERS: U-ApiKey: YOUR_API_KEY_HERE

{
"timestamp":"2012-03-15T16:13:14",
"value":294.34
}

まず自分のデバイスとセンサーをyeelinkで登録して、device_idとsensor_idを確認します。
次は自分のAPIのキーを調べて、リクエストのヘッダーに置きます。
リクエストの内容は時間と数値です。
```

# 2.5 program

```
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
4 // TMP sensor connect to A0
5 #define PIN_TMP 0
7 int analogValue = 0;
8 float voltage = 0;
9 float celsius = 0;
10 int serialCmd;
11
12 LiquidCrystal_I2C lcd(0x27,16,2);
13
14 void setup() {
15
     Serial.begin(9600);
16
     lcd.init();
17
     // lcd.backlight();
     lcd.print("ALL CLEAR!");
18
19 }
20
21 void loop() {
22
     analogValue = analogRead(PIN_TMP);
23
     voltage = analogValue * 5000.0 / 1024;
24
     celsius = (voltage - 500) / 10;
25
26
     // show temperature on LCD
27
     lcd.clear();
28
     lcd.print(celsius);
29
30
     // send to serial if necessary
31
     if(Serial.available() && Serial.read() == 71 /* G */) {
32
       Serial.println(celsius);
33
34
     delay(1000);
35 }
```

source: temperature-collector.py

```
def loop(s):
15
16
17
         s.writelines(['G']) # send serial command to get current temperature
         c = s.readline().strip()
         if not c:
         now = datetime.datetime.now().isoformat()
         print '[INFO]', now, c
21
22
23
24
25
26
27
28
29
30
          if send_sensor_data(now, c):
              GPIO.output(PIN_LED_PASSED, GPIO.HIGH)
         else:
              GPIO.output(PIN_LED_FAILED, GPIO.HIGH)
         sleep(1)
         GPIO.output(PIN_LED_PASSED, GPIO.LOW)
         GPIO.output(PIN_LED_FAILED, GPIO.LOW)
         sleep(10) # sleep 10 seconds for yeelink limit(> 10s after previous request)
     def send_sensor_data(timestamp, value):
    payload = '{"timestamp":"' + timestamp + '","value":' + value + '}'
    headers = {'U-ApiKey': API_KEY}
31
32
33
34
35
         conn = httplib.HTTPConnection(HOST)
         conn.request('POST', URI, payload, headers)
         resp = conn.getresponse()
37
         conn.close()
38
         if resp.status == 200:
39
              return True
40
              print '[WARN] failed to send sensor data,', resp.reason
              return False
43
                 == '__main_
         GPIO.setmode(GPIO.BCM)
         GPIO.setup(PIN_LED_PASSED, GPIO.OUT)
47
         GPIO.setup(PIN_LED_FAILED, GPIO.OUT)
49
         s = serial.Serial('/dev/ttyACM0', 9600, timeout = 1)
50
51
52
53
54
         s.open()
         print '[INFO] sensor data collector start'
          try:
              while True:
                  loop(s)
         except KeyboardInterrupt:
56
              s.close()
              GPIO.cleanup()
```

APIキーなどの情報があるので、完全のコードが表示されていません。