

A new horizontal seismograph system for school use employing “Arduino” and “Processing”

Yoshio Okamoto, Osaka-Kyoiku University

(yossi@cc.osaka-kyoiku.ac.jp)

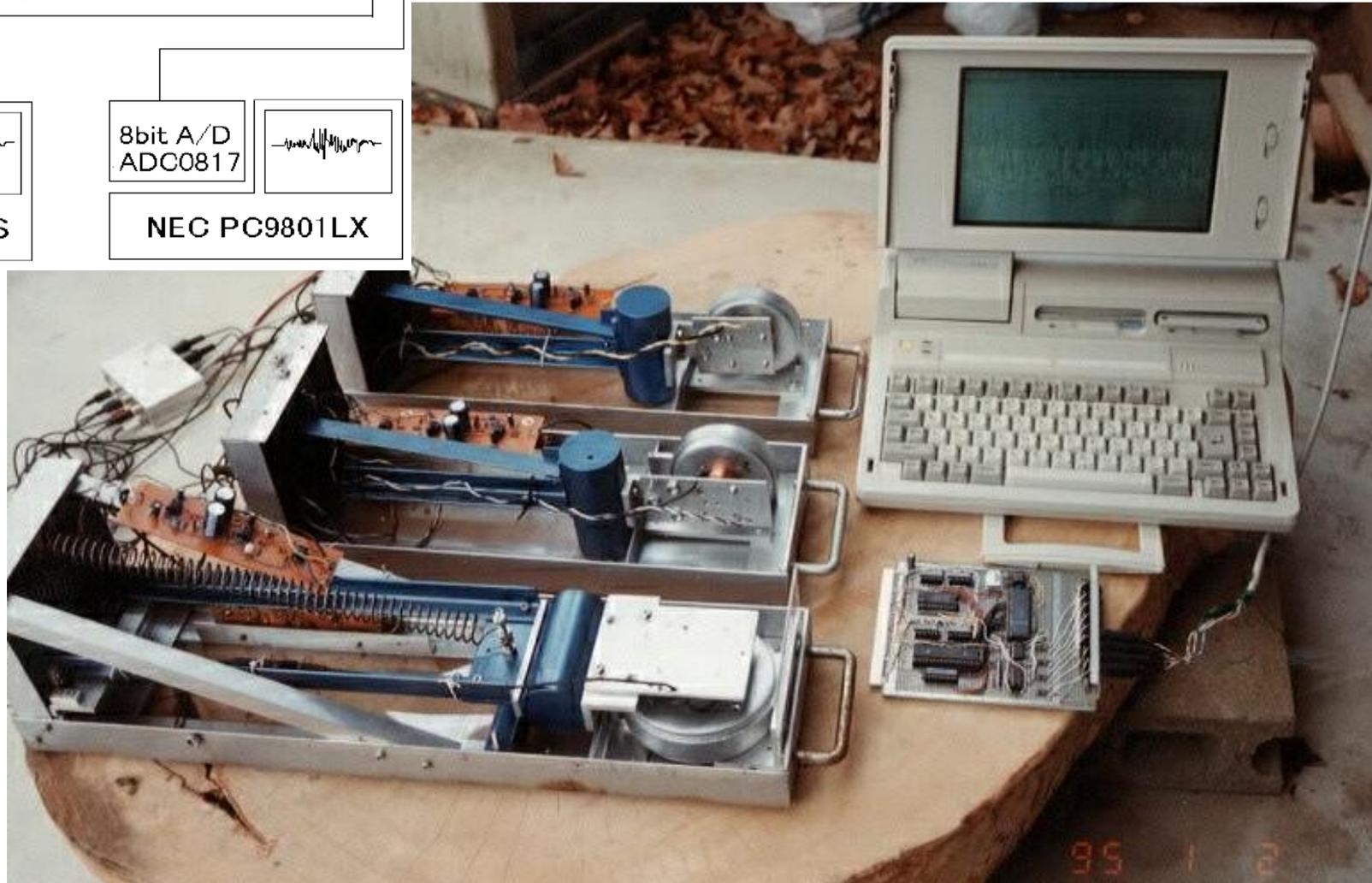
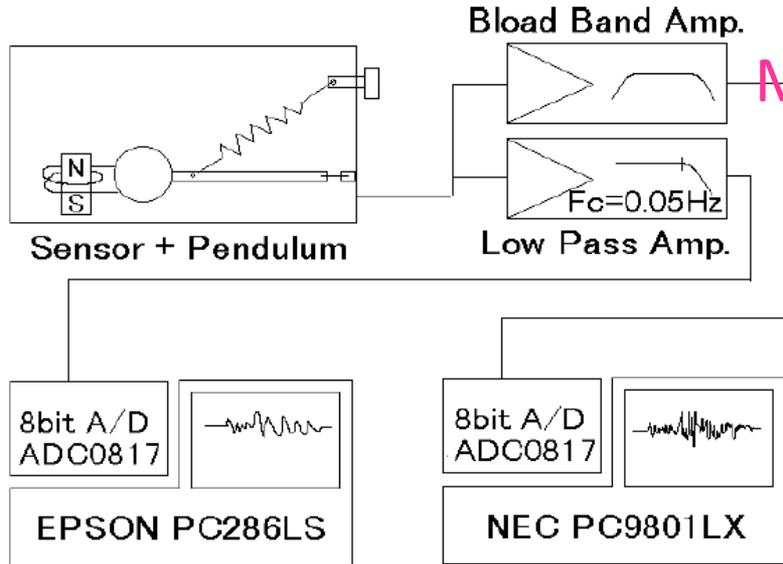
Akihiko Ito, Utsunomiya University

(ito@cc.utsunomiya-u.ac.jp)

GeoSciEd VII Hyderabad 2014

From Yoshio OKAMOTO: A Handmade Seismograph System and its Seismograms
-Make Your Own Seismograph!- abstract for GeoSciEdu III in Sydney, 2000

My first seismometers for education:
Complicated metal handicraft
and a legacy hard-ware



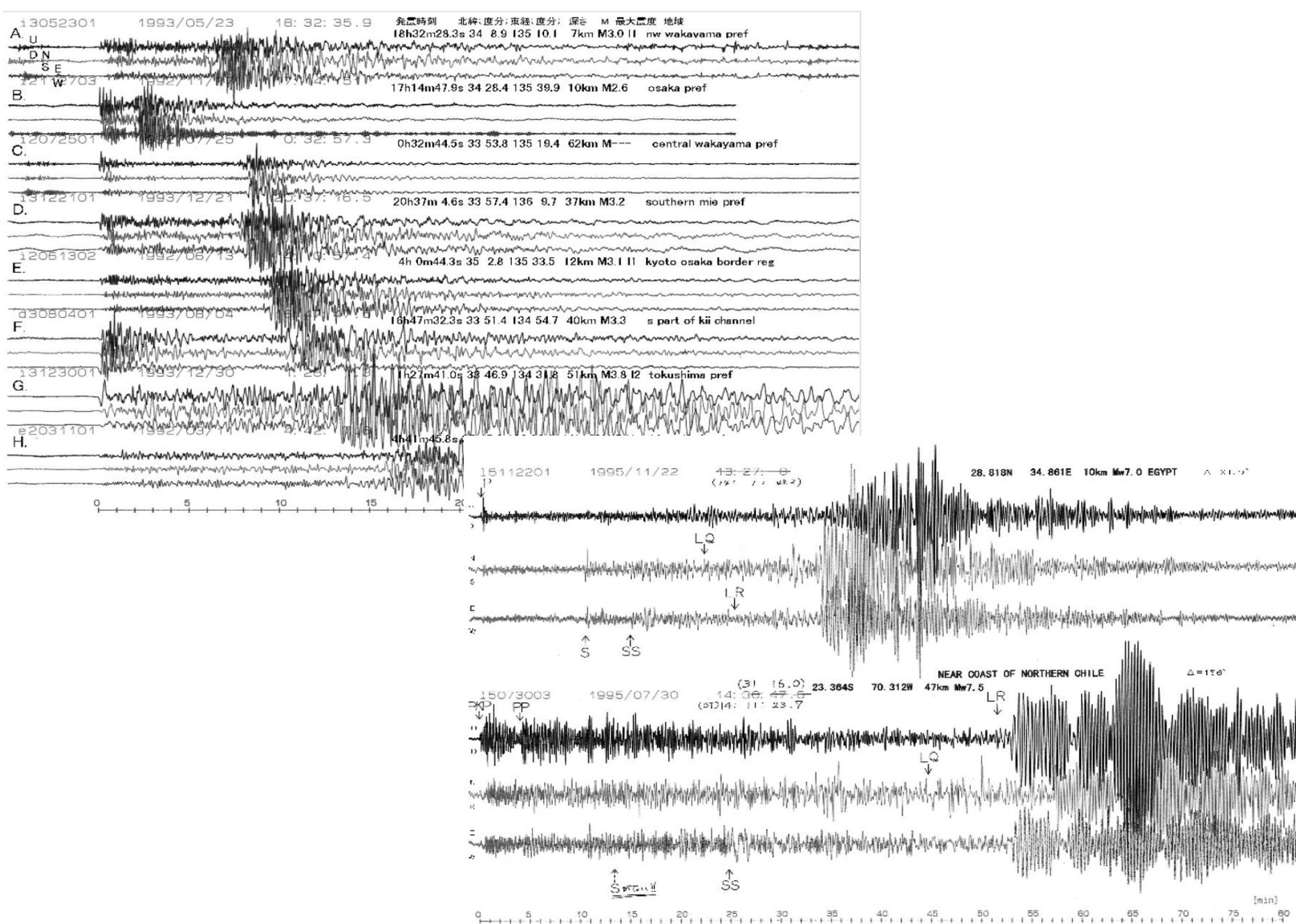


Fig.5 Seismograms of foreign earthquakes (Upper: Egypt Mw7.0 Lower: Chile Mw7.5).

My second seismometer (vertical comp.) and a new recording system

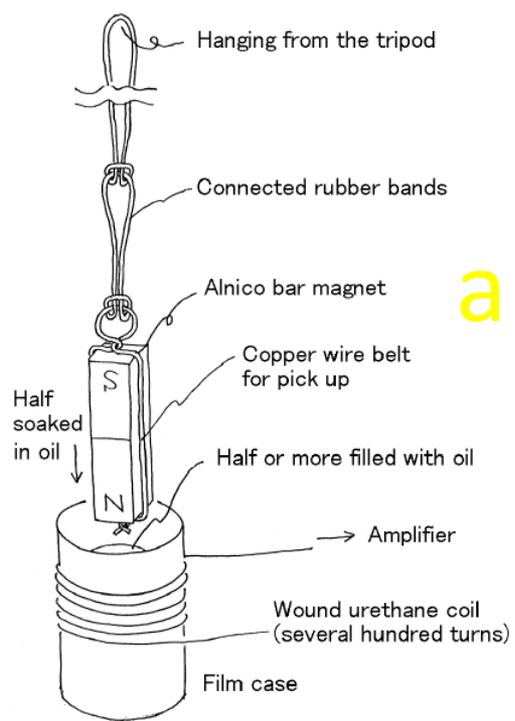


Fig 6. Film case sensor



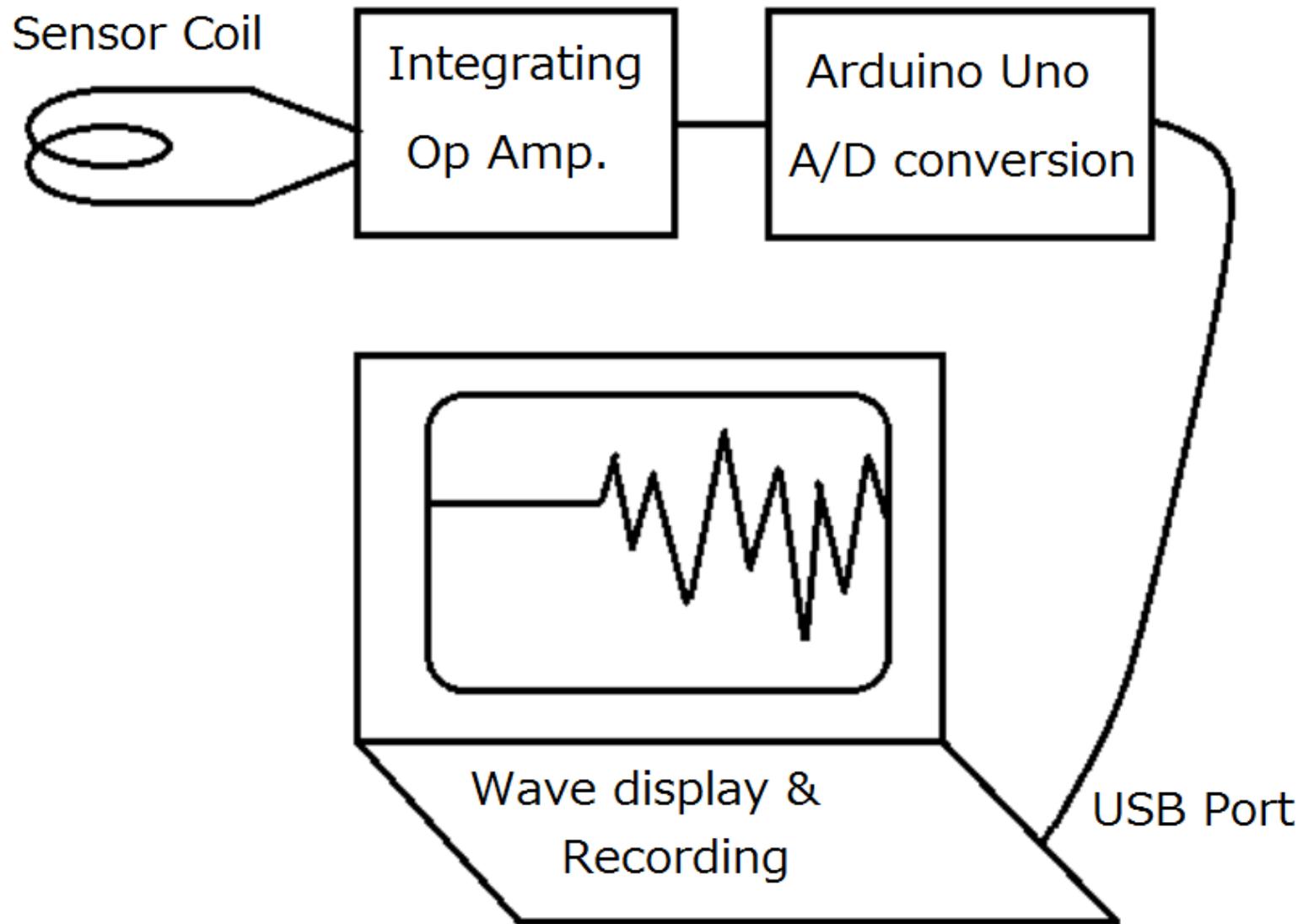
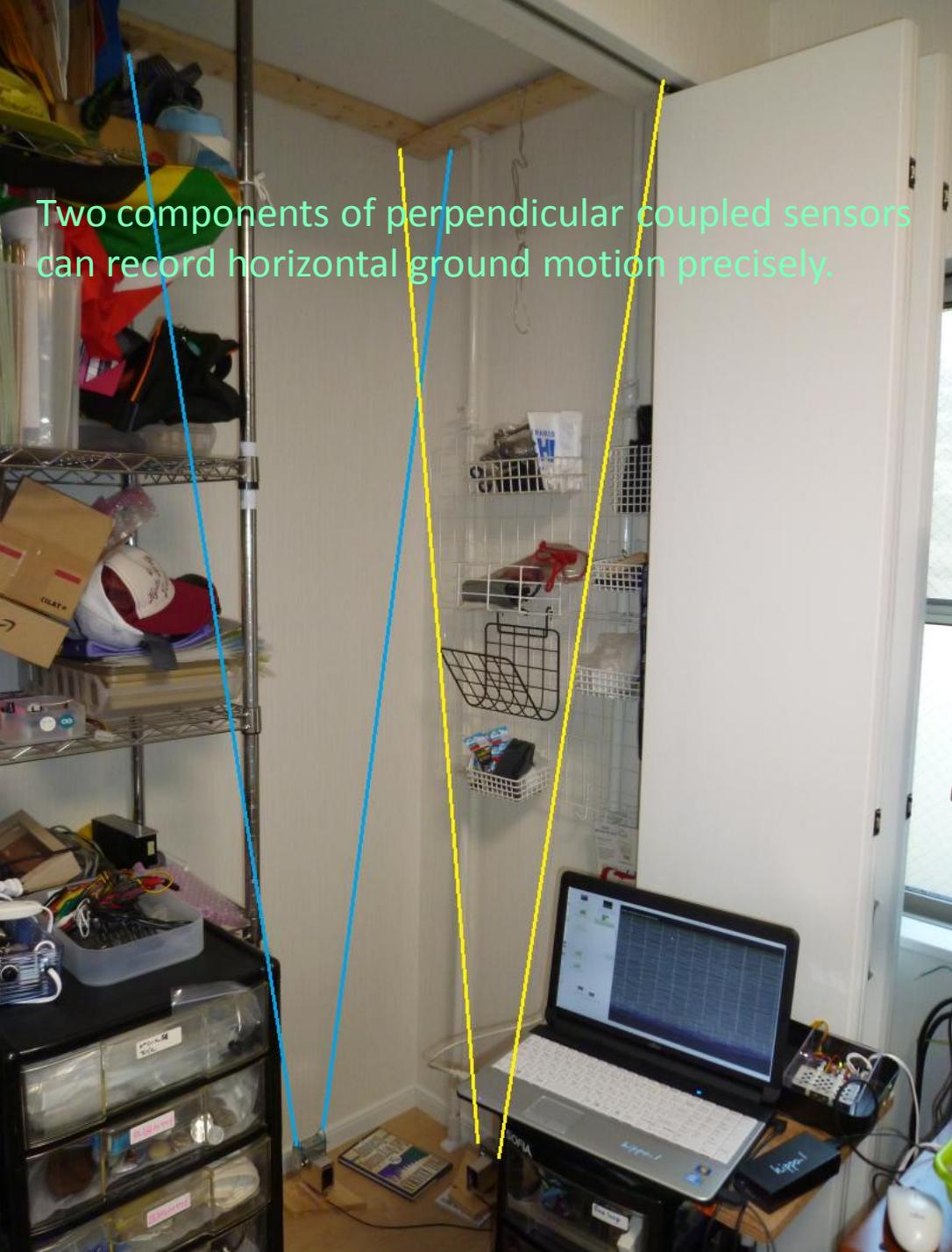


Fig.1 System over view



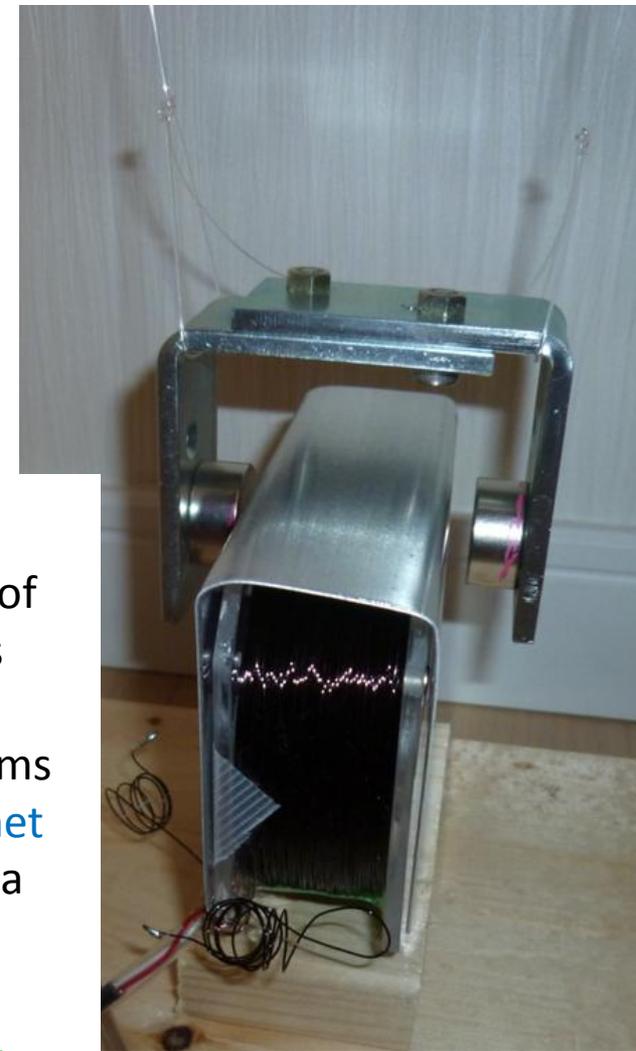
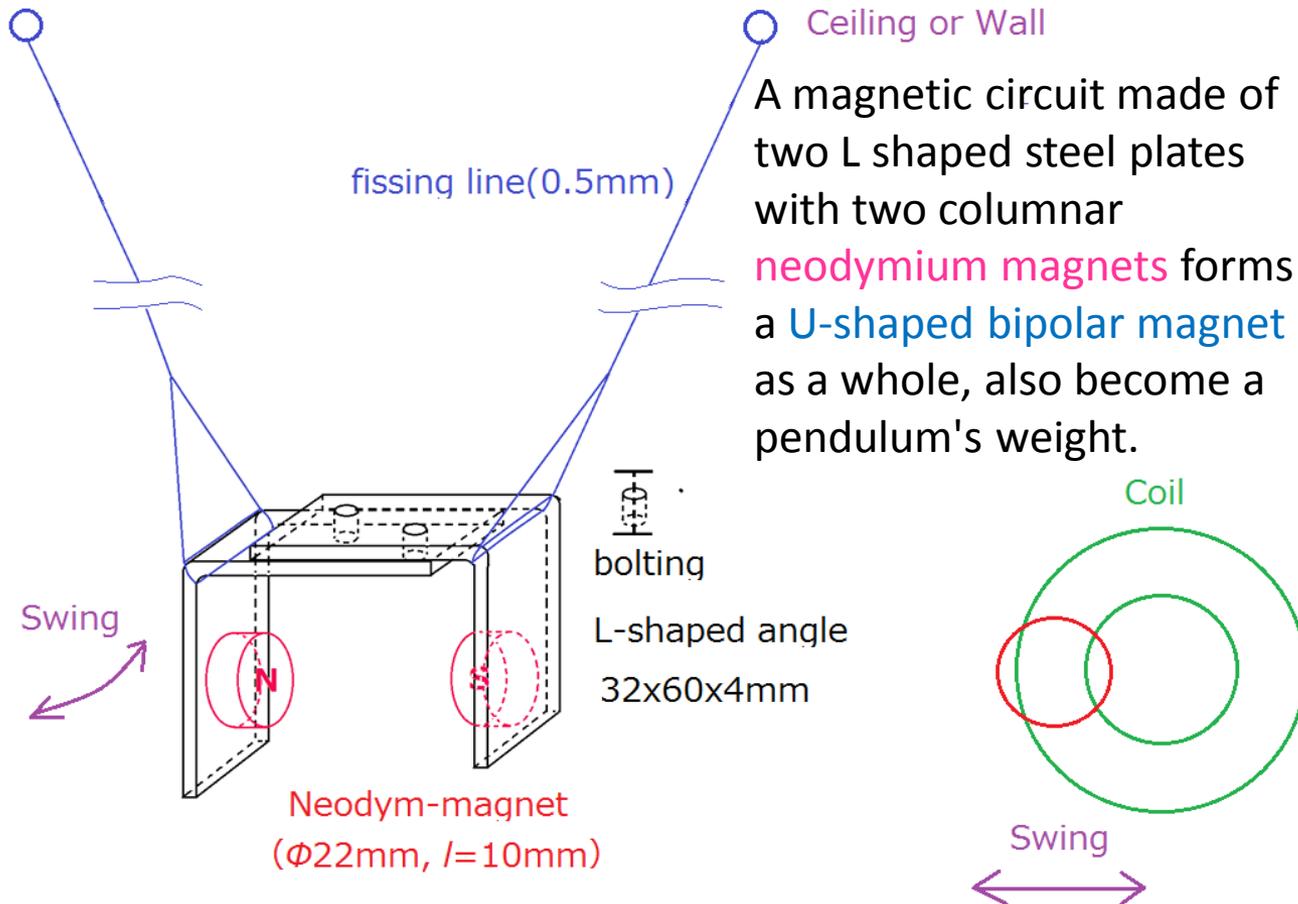
New seismometers
and PC at my room
(2F, wooden house)

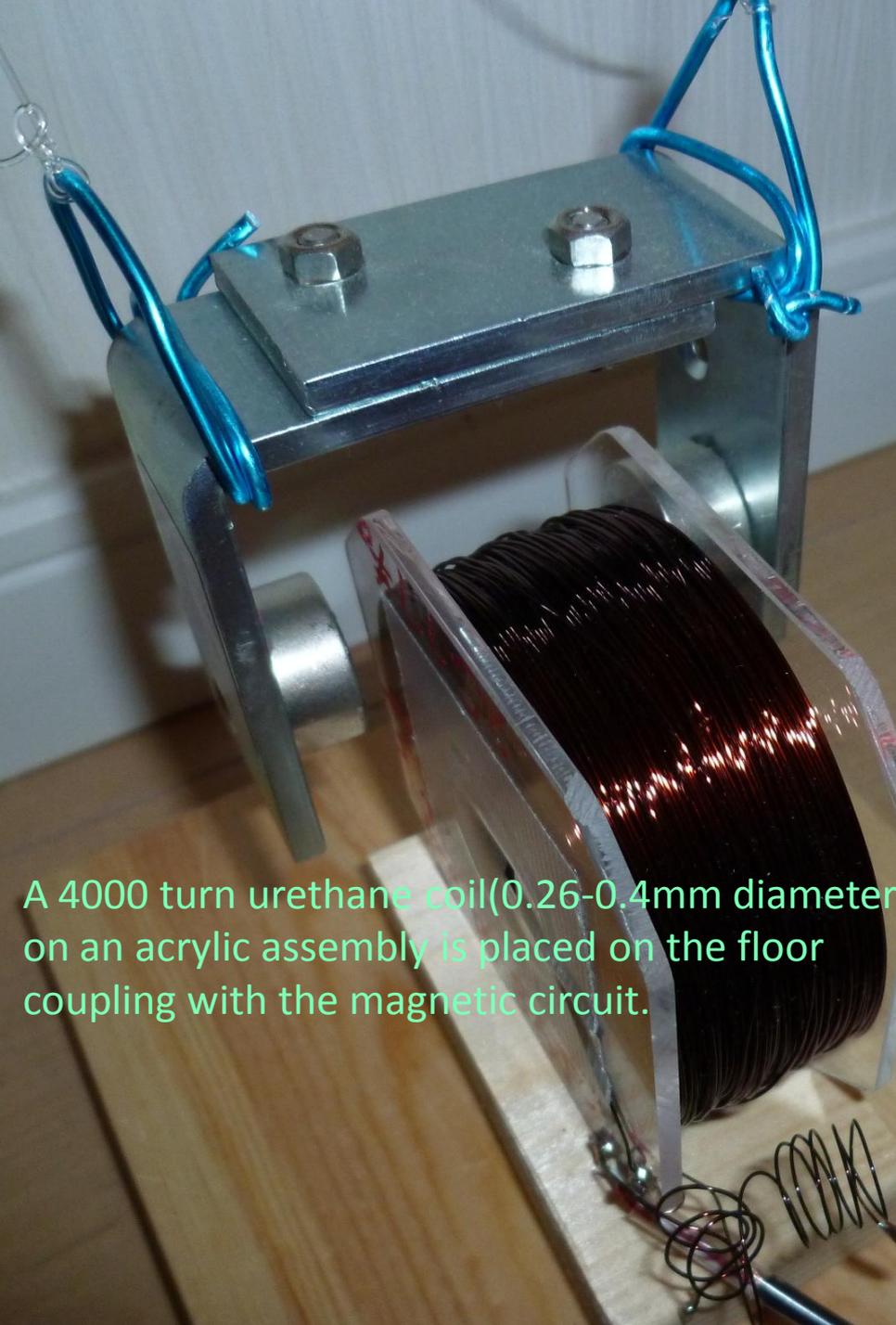


Two components of perpendicular coupled sensors
can record horizontal ground motion precisely.

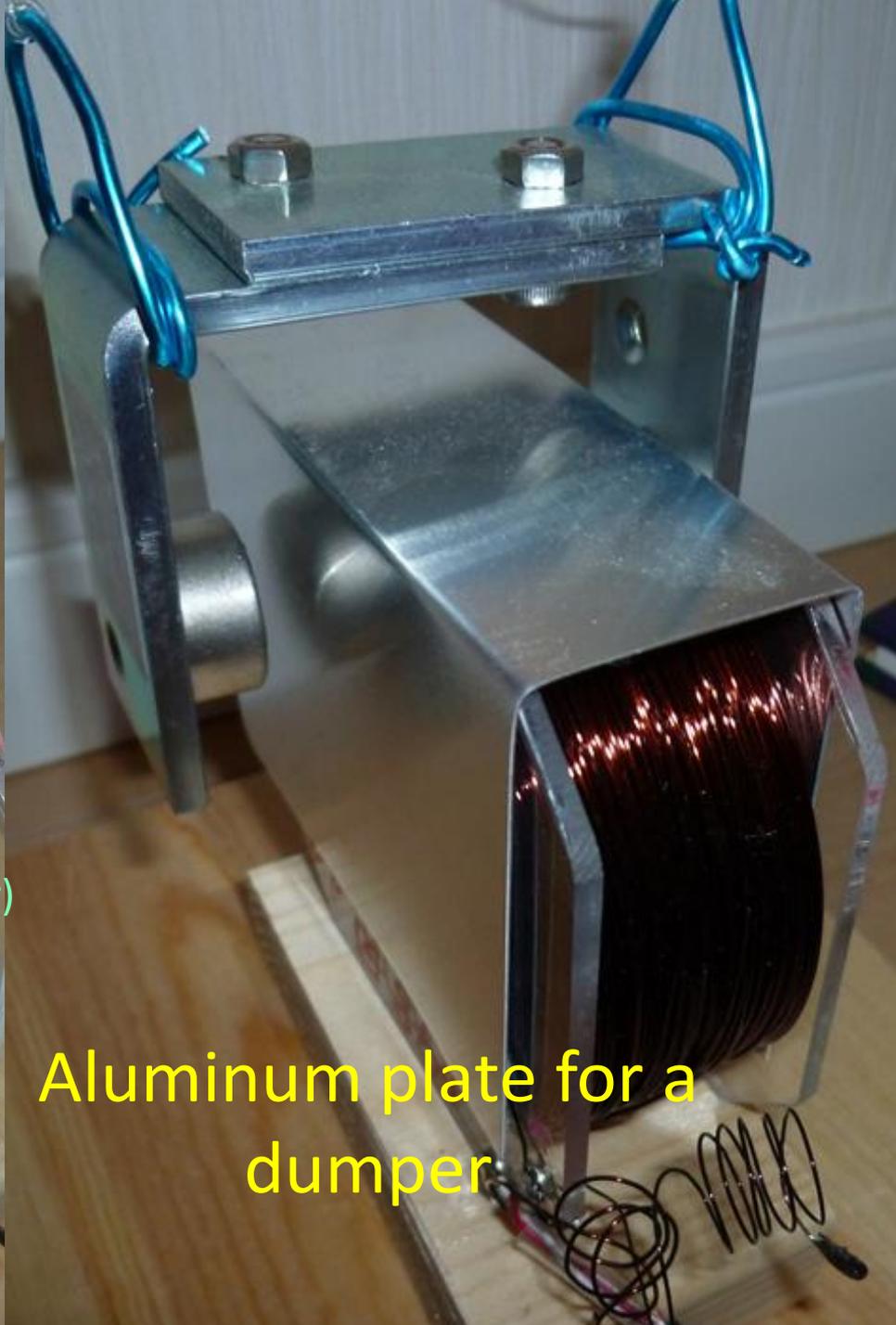
A simple **bifilar suspension** is used for the horizontal pendulum.

The aluminum plate covering coil is used as a “dumper”





A 4000 turn urethane coil(0.26-0.4mm diameter) on an acrylic assembly is placed on the floor coupling with the magnetic circuit.



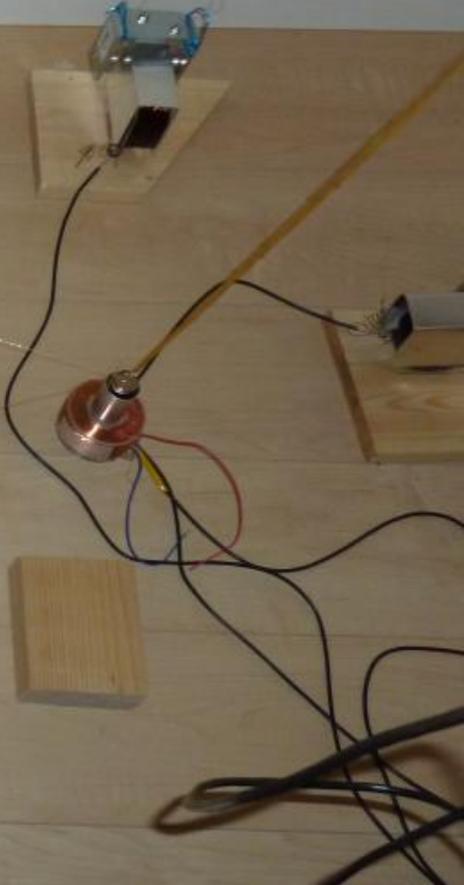
Aluminum plate for a dumper

Several vertical pendulums using same sensor are now testing ---.





FREE BOX A4R



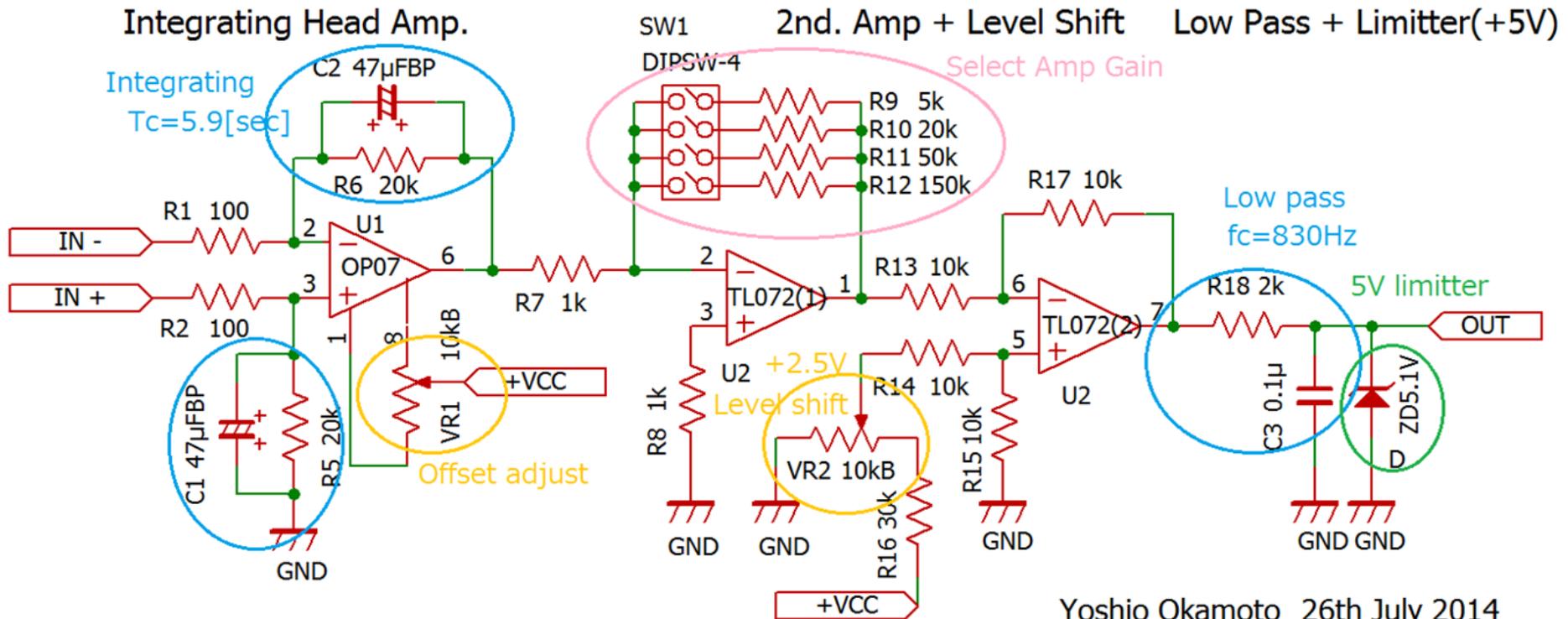
Amplifier circuit (parts: 10USD)

1st amp. gain

2nd amp. gain

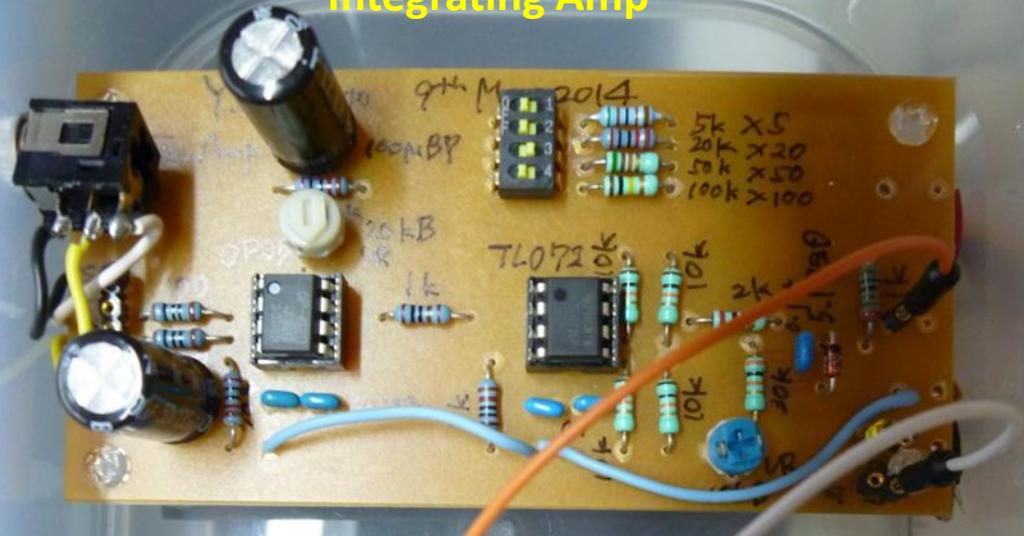
x200(Tc=6sec, -6dB/oct)

x20~50



Integrating Amp. for 1ch. and A/D converter: “Arduino Uno” (30 USD)

Integrating Amp



The signal generated in the coil, which is proportional to ground velocity, is at first amplified and integrated with OP-amp circuit, then digitalized with Arduino(10 bit A/D micro-chip, 30 USD), and finally guided into a PC via a USB port.



Arduino Uno

```
const int accel_pin[] = { 0, 1, 2 };
```

```
int accel_val[] = { 0, 0, 0 };
```

```
void setup() {  
  Serial.begin( 9600 );  
}
```

```
void loop() {  
  byte buffer[2];
```

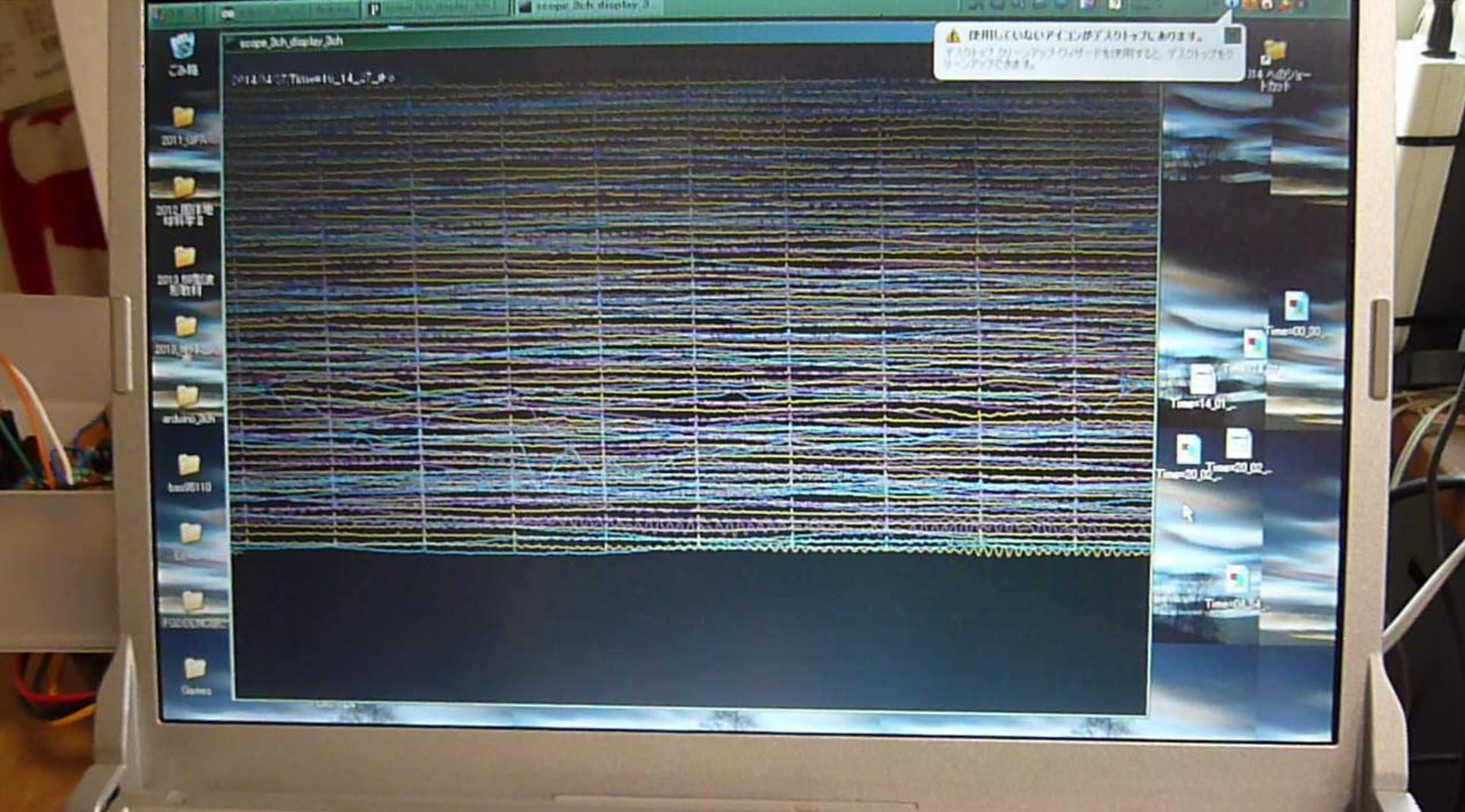
```
  // AD convert for 3-ch  
  for ( int i = 0; i < 3; i++ ) {  
    accel_val[i] = analogRead( accel_pin[i] ); // read pin[i]  
  }
```

```
  // Data transfer: if processing send one byte character * to Arduino  
  if ( Serial.available() > 0 ) {  
    for ( int i = 0; i < 3; i++ ) {  
      buffer[0] = byte( accel_val[i] );  
      buffer[1] = byte( accel_val[i] >> 8 );  
      Serial.write( buffer, 2 );  
    }  
    Serial.read(); // take off * character  
  }  
}
```

"Arduino IDE" for
signal processing

Recording Program: Using "Processing": JAVA based programming language

The PC displays signals as wave lines at real time and can save the signals both as a screenshot and/or a text data file for later analysis. The saved data are 10 minutes length each and their sampling rate is adjusted at 100Hz. Now, the system extended for three channel inputs.

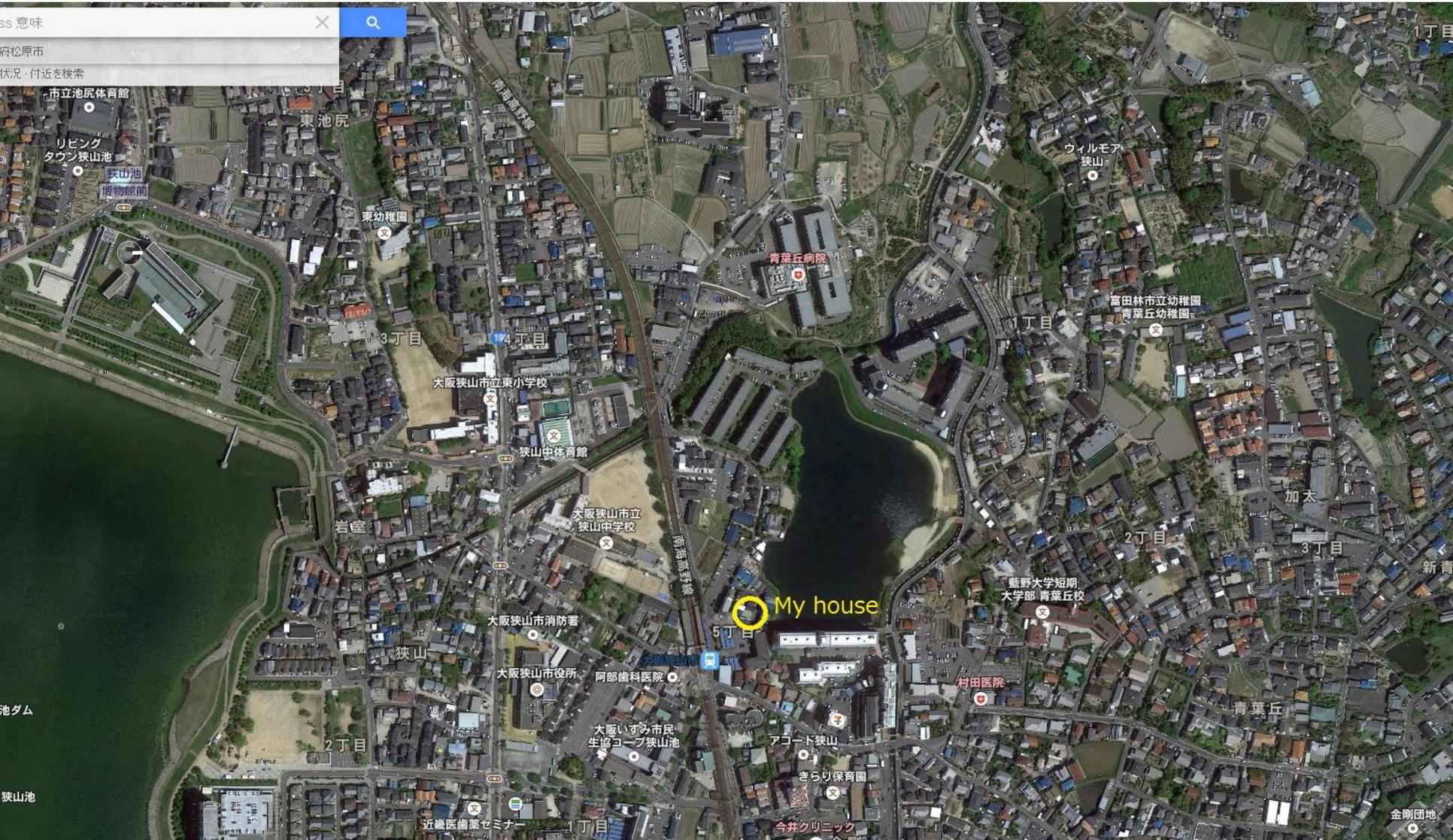


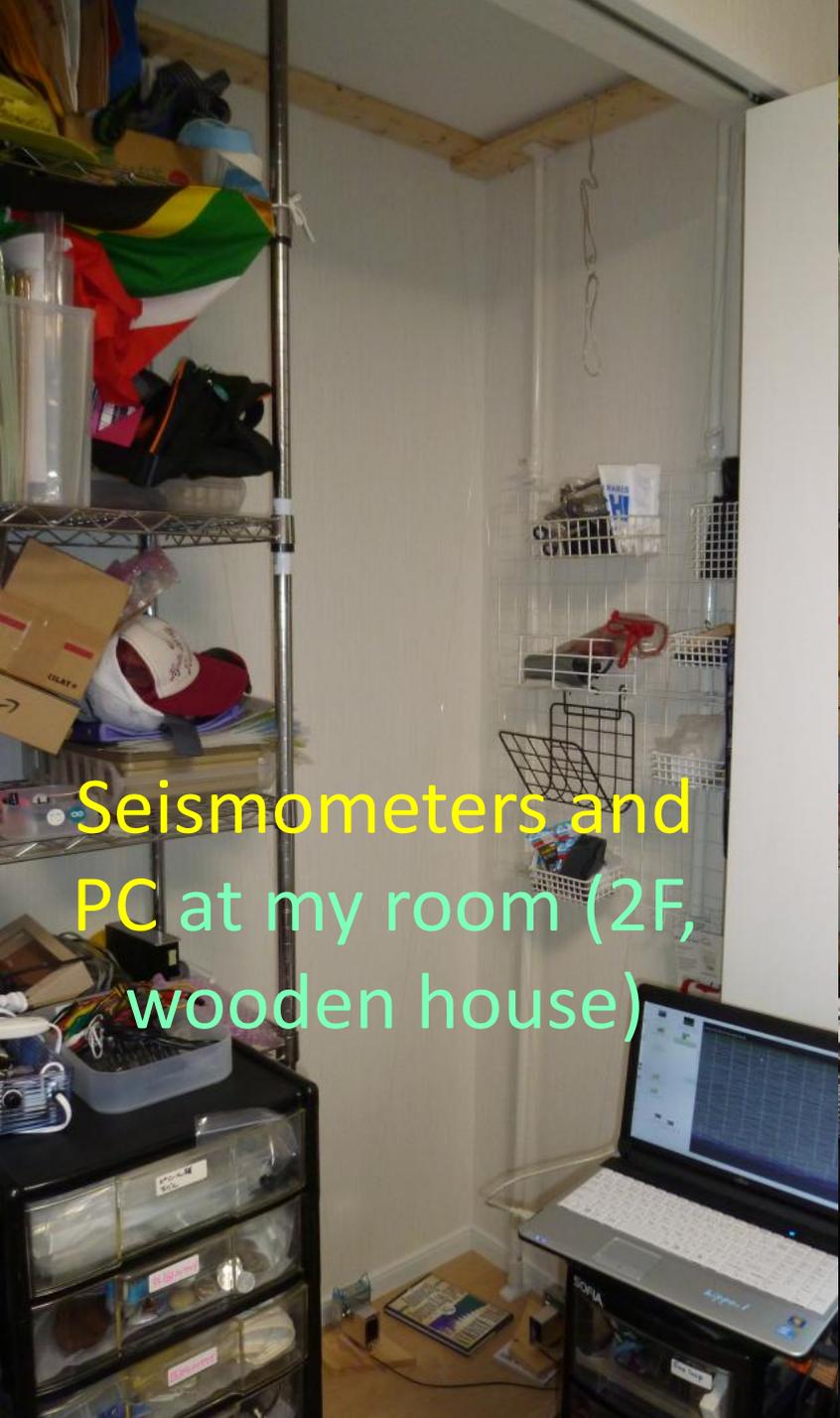
1ch-Display Program employing "Processing"

```
seism_demo.pde (~/Desktop/seism_demo) - pluma
ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)
開く 保存 元に戻す 印刷 検索
seism_demo.pde
1 // Example 19-8: Reading from serial port orig. by Daniel Shiffman
2 // modified by Yoshio Okamoto 22th Nov.2013, 17th Mar.2014
3
4 import processing.serial.*;
5
6 Serial port; // The serial port object
7
8 PrintWriter output;
9
10 int i = 0;
11 int t = 0;
12 int tt = 0;
13 int ttt = 0;
14 int ch = 0;
15 float x[] = { 0,0,0 };
16 float y[] = { 0,0,0 };
17 int base[] = { -400, -400, -400 };
18 //int base[] = { 0, 0, 0 };
19 float px[] = { 0,0,0 };
20 float py[] = { 0,0,0 };
21 int data[] = { 0,0,0 };
22 int data_1[];
23 int data_2[];
24 int data_3[];
25 int s = 0;
26 int m = 0;
27 int ps = 0;
28 int ts = 0;
29 int tmax = 1000 * 60;
30 int sflg=0;
31
32 String datastr;
33 String datetimestr;
34 String format;
35
36 void setup() {
37   size(1000,720); // display size
38   strokeWeight( 1 ); // line width
39   background( 0 ); // clear display
40   port = new Serial(this, Serial.list()[0], 9600); // set up port for the first serial port
41   frameRate(100); // sampling rate/sec
42   // stroke(255);
43   // smooth();
44 }
45
46 void draw() {
47
48 // initialize
49 if (sflg==0){
50   delay(1000);
51   port.write( '*' );
52   background(0);
53   t=0;
54   tt=0;
55   sflg=1;
56   port.write( '*' );
57 }
58
59 // file init.
60 if (t == 0){
61   datetimestr = year()+"/"+month(,2)+"/"+day(,2)+"/Time="+hour(,2) + ":" + minute(,2) + ":" +
62   nf(second(,2)+"_IFS"; // for linux
63   //datetimestr = year()+"/"+month(,2)+"/"+day(,2)+"/Time="+hour(,2) + ":" + minute(,2) + ":" +
64   nf(second(,2)+"_IFS"; // for Windows
65   output = createWriter(datetimestr+".dat");
66   text( datetimestr,10,35);
67   port.write( '*' );
68 }
69
70 // time mark
71 s = second();
72 if (s == ps)
73   tm = 0;
74 else if (s % 10 == 0)
75   tm = 8;
76 else
77   tm = 5;
78 ps = s;
79
80 // display ch
81 ch=0;
82 stroke (50, 190, 255); // for 1ch with light blue color
83
84 x[ch] = tt;
85 y[ch] = data[ch] + base[ch] + ttt*10 - tm;
86
87 // display signal
88 if ((t==0) || (tt == 0))
89   point (x[ch], y[ch]);
90 else
91   line (px[ch], py[ch], x[ch], y[ch]);
92
93 px[ch] = x[ch];
94 py[ch] = y[ch];
95
96 // data to text file
97 format = String.format("%04d %04d %04d", data[0],data[1],data[2]);
98 output.println(format);
99
100 tt = t % (width);
101 ttt = int (t/width);
102 t = t+1;
103
104 if (t>tmax){
105   t = 0;
106   tt = 0;
107 }
108 // save window
109 save( datetimestr+".png" );
110
111 output.flush(); // write buffer to file
112 output.close(); // close file
113
114 background(0);
115 }
116
117 // key events
118 void keyPressed() {
119   if ( key == 's' ) {
120     port.write( '*' );
121     background(0);
122     t=0;
123     tt=0;
124     text( year()+"/"+month(,2)+"/"+day(,2)+"/Time="+hour(,2)+":"+minute(,2)+":"+second(,2),10,35);
125   }
126   if ( key == 'v' ) {
127     save( year()+"/"+month(,2)+"/"+day(,2)+"/Time="+hour(,2)+":"+minute(,2)+":"+second(,2)
128     +"FS"+".png" ); // for linux
129     //save( year()+"/"+month(,2)+"/"+day(,2)+"/Time="+hour(,2)+"_" +minute(,2)+"_" +second(,2)
130     +"FS"+".png" ); // for Windows
131   }
132   if ( key == 'c' ) {
133     ch = ch + 1;
134     if (ch > 2)
135       ch=0;
136   }
137 }
138 // for 3ch input
139 void serialEvent( Serial p ) {
140   if ( port.available() > 1 ) {
141     for ( i = 0; i < 1; i++ ) {
142       data[i] = port.read() + port.read() * 256 ;
143     }
144     port.write( '*' );
145   }
146 }
147
148
149
150
151
```

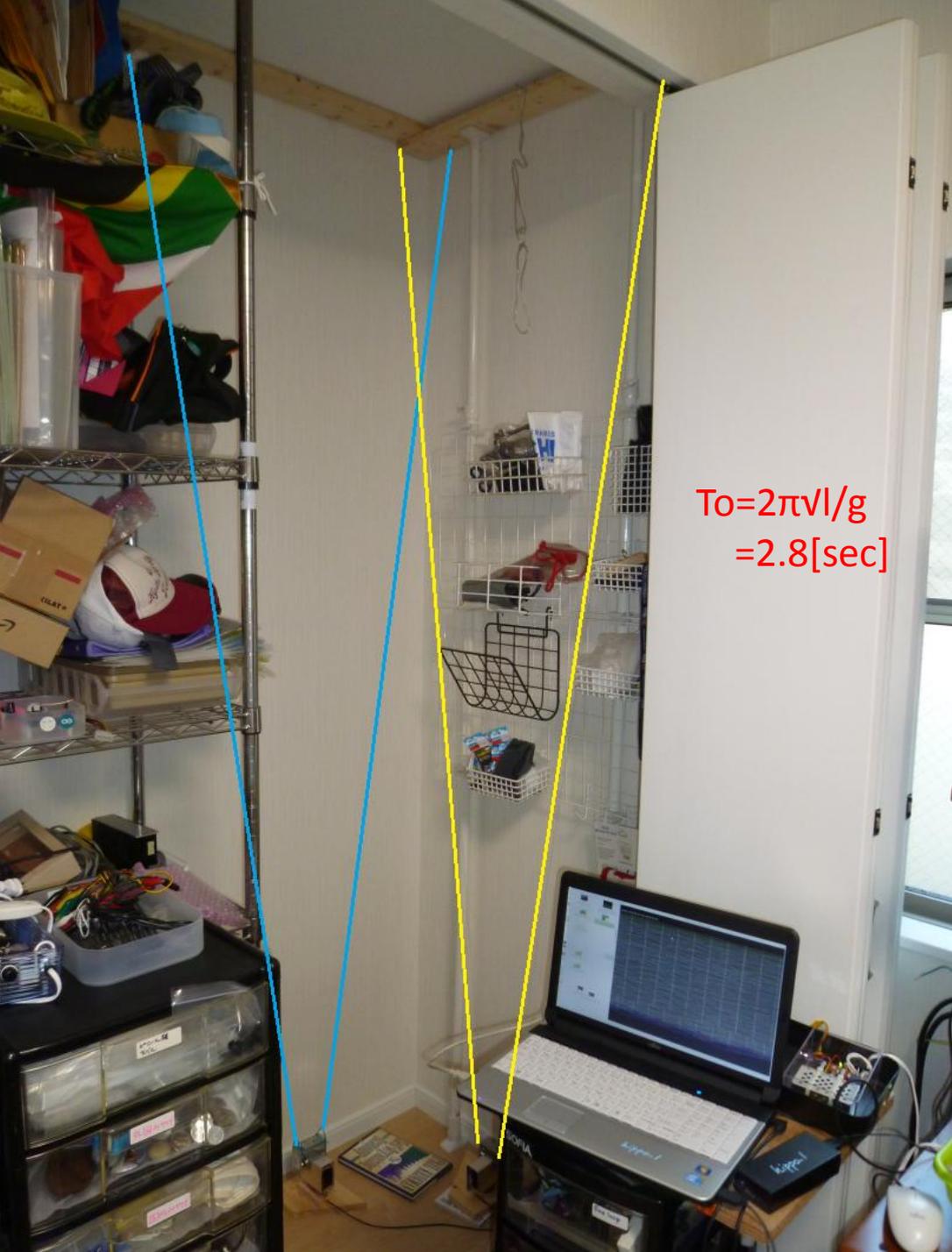
```
*編集中のドキュメント 1 - pluma
ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)
開く 保存 元に戻す 印刷 検索
*編集中のドキュメント 1
77
78 // 3ch signal display (1ch only)
79
80 // display ch
81 ch=0;
82 stroke (50, 190, 255); // for 1ch with light blue color
83
84 x[ch] = tt;
85 y[ch] = data[ch] + base[ch] + ttt*10 - tm;
86
87 // display signal
88 if ((t==0) || (tt == 0))
89   point (x[ch], y[ch]);
90 else
91   line (px[ch], py[ch], x[ch], y[ch]);
92
93 px[ch] = x[ch];
94 py[ch] = y[ch];
95
96 // data to text file
97 format = String.format("%04d %04d %04d", data[0],data[1],data[2]);
98 output.println(format);
99
100 tt = t % (width);
101 ttt = int (t/width);
102 t = t+1;
103
104 if (t>tmax){
105   t = 0;
106   tt = 0;
107 }
108 // save window
109 save( datetimestr+".png" );
110
111 output.flush(); // write buffer to file
112 output.close(); // close file
113
114 background(0);
115 }
116
117 // key events
118 void keyPressed() {
119   if ( key == 's' ) {
120     port.write( '*' );
121     background(0);
122     t=0;
123     tt=0;
124     text( year()+"/"+month(,2)+"/"+day(,2)+"/Time="+hour(,2)+":"+minute(,2)+":"+second(,2),10,35);
125   }
126   if ( key == 'v' ) {
127     save( year()+"/"+month(,2)+"/"+day(,2)+"/Time="+hour(,2)+":"+minute(,2)+":"+second(,2)
128     +"FS"+".png" ); // for linux
129     //save( year()+"/"+month(,2)+"/"+day(,2)+"/Time="+hour(,2)+"_" +minute(,2)+"_" +second(,2)
130     +"FS"+".png" ); // for Windows
131   }
132   if ( key == 'c' ) {
133     ch = ch + 1;
134     if (ch > 2)
135       ch=0;
136   }
137 }
138 // for 3ch input
139 void serialEvent( Serial p ) {
140   if ( port.available() > 1 ) {
141     for ( i = 0; i < 1; i++ ) {
142       data[i] = port.read() + port.read() * 256 ;
143     }
144     port.write( '*' );
145   }
146 }
147
148
149
150
151
```

Test observation at my house





Seismometers and
PC at my room (2F,
wooden house)



$$T_0 = 2\pi\sqrt{l/g}$$
$$= 2.8[\text{sec}]$$

A felt natural
earthquake at my house

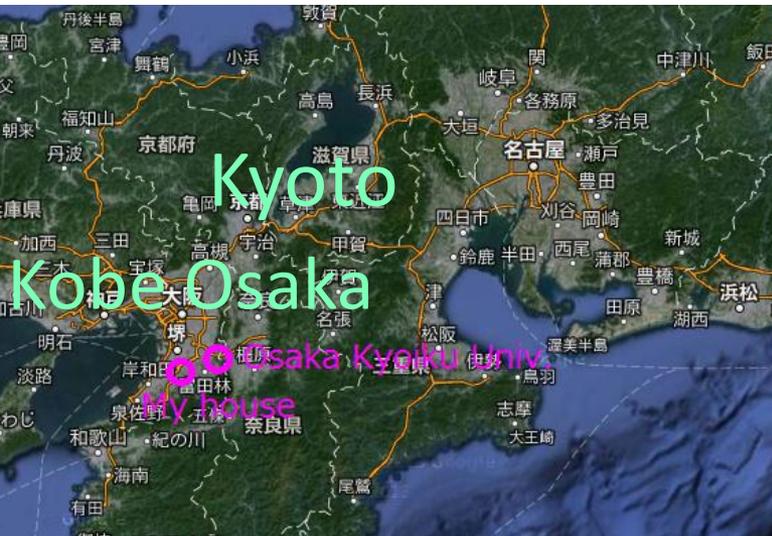
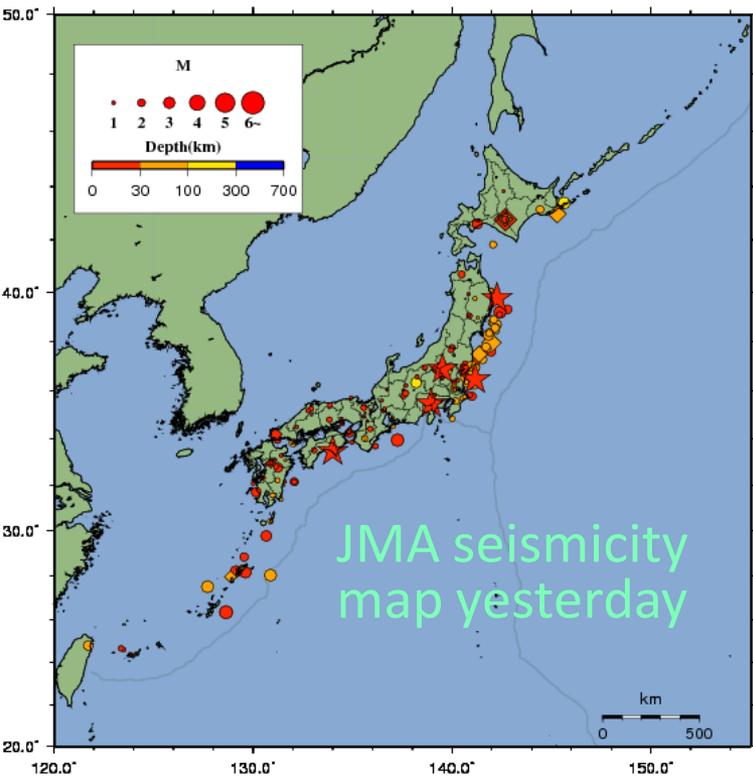
Real-time display is
saved as a screen shot.

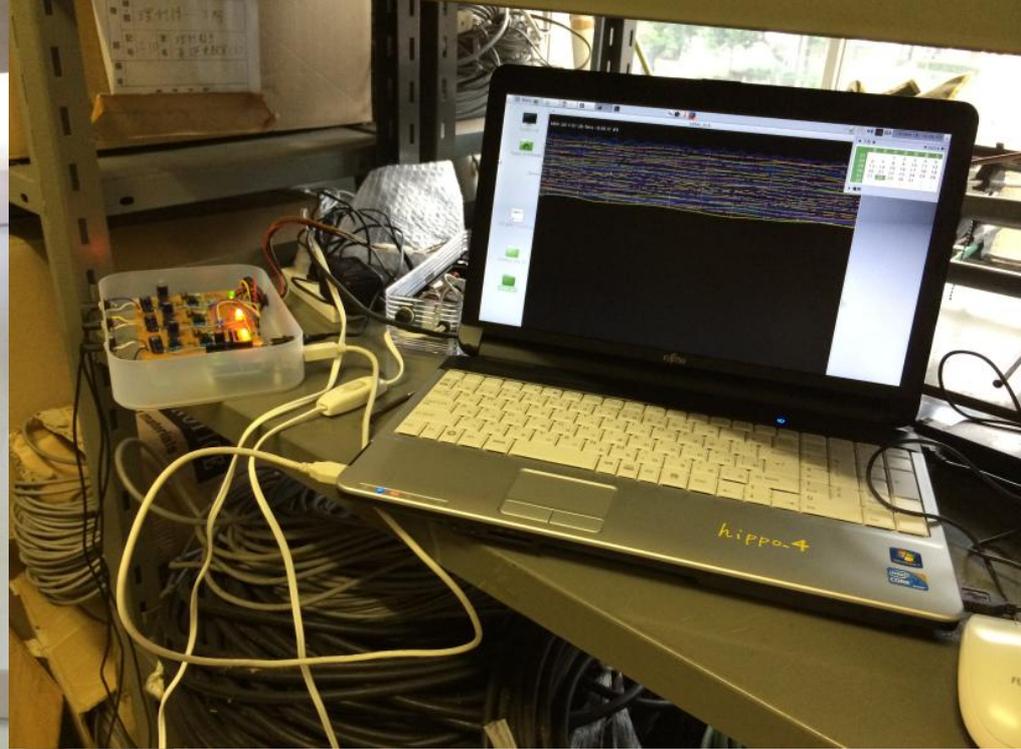


Soft-ware

- The PC soft-wares used here are written by "Arduino IDE" for signal processing and "Processing" for signal display and data recording, both freely downloadable from their web sites. Also they can run on Windows, Mac OS, Linux or even Android tablet. The Processing language is based on JAVA and more simplified for programing beginners. The total cost is around 100 USD and easy to assemble at school or home use.

2014/09/04 00:00 ~ 2014/09/05 09:10



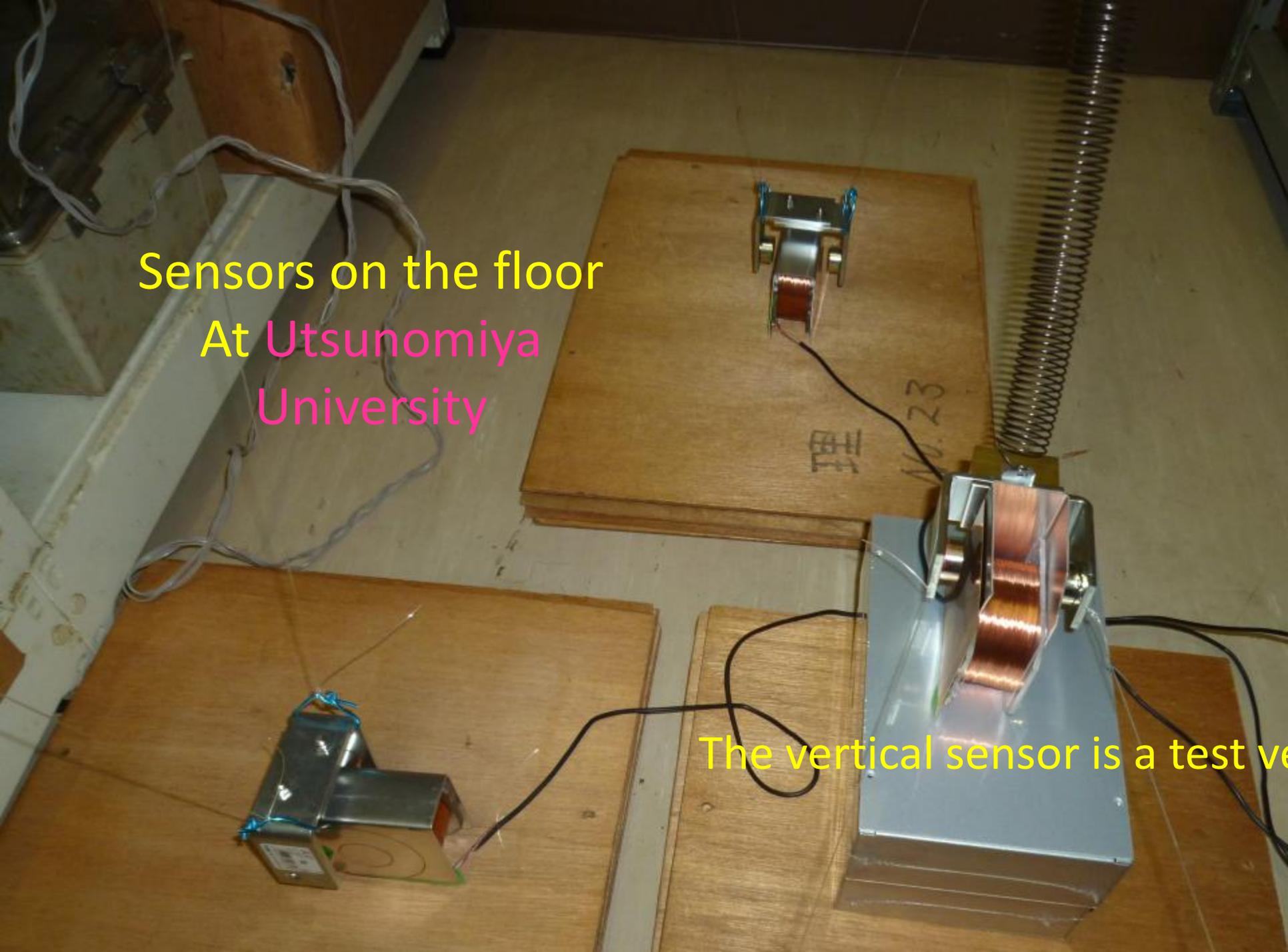


Seismometers and a recording PC
are set on the floor and the shelf:
At **Utsunomiya University**

The PC uses Linux as its OS.

Sensors on the floor
At Utsunomiya
University

The vertical sensor is a test ve



A Seismogram at
Utsunomiya University
(in Kanto plain)

Microseisms are prominent by
thick alluvials and Kanto loams ~ 2km

072-976-3211



子短期
大寺附属幼稚園



大阪教育大学
附属図書館

大阪教育大学
教員養成・学生支援

Osaka Kyoiku University

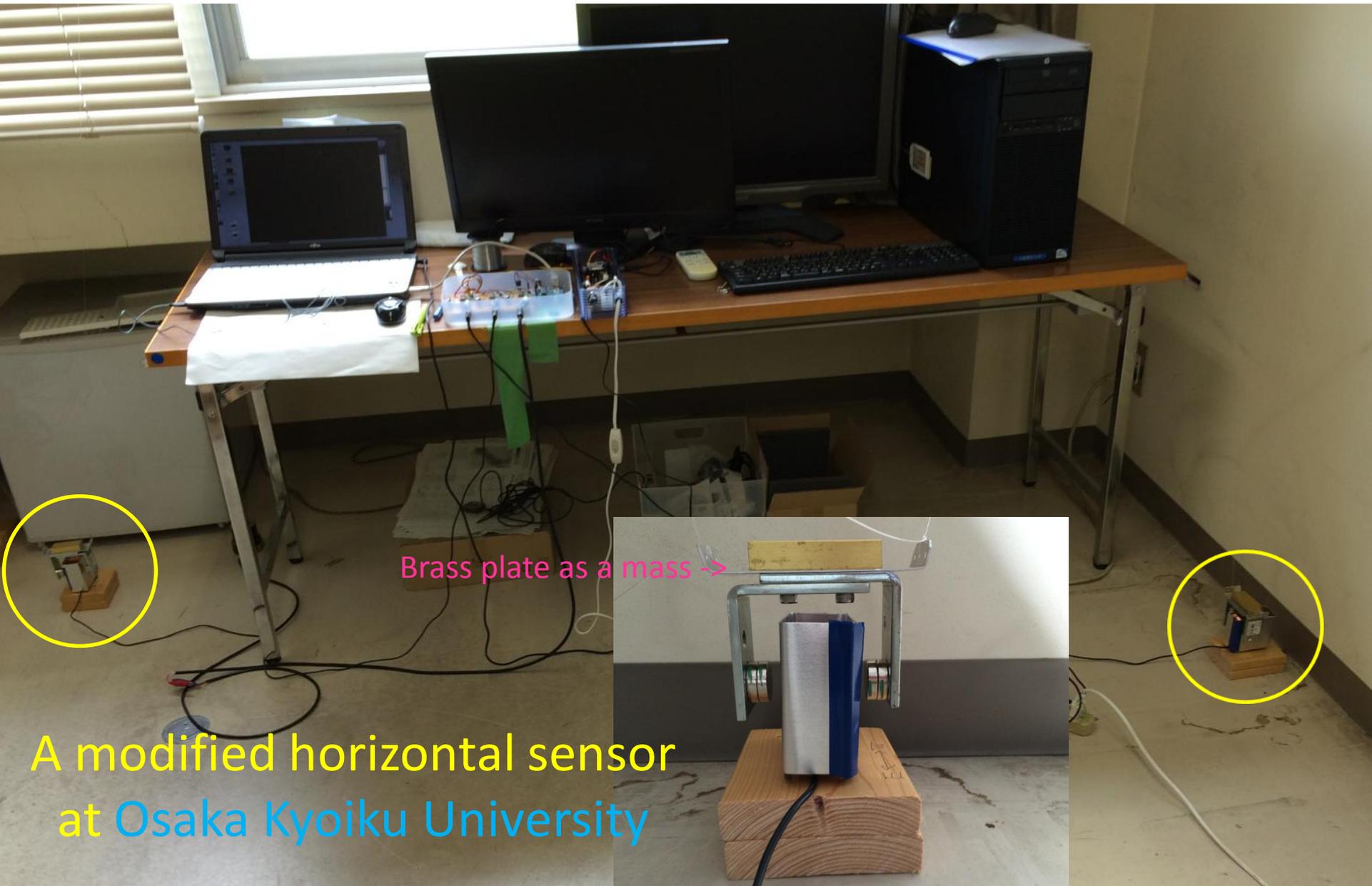
クレーブビル
スポーツ公園

はびきの中央公園

A photograph of a computer workstation in a room. On the left wall, there is a white air conditioner. A window with white horizontal blinds is in the center, with grey curtains on either side. A desk is positioned in front of the window, holding a laptop, two monitors, a keyboard, and a mouse. A black tower PC is on the right side of the desk. A small step ladder is in front of the desk. On the floor, there are several cardboard boxes, a metal trolley, and various cables. The room appears to be a laboratory or office.

<- Air conditioner

Seismometers and PC at
Osaka Kyoiku University
(on a hill top)



Brass plate as a mass →

A modified horizontal sensor
at Osaka Kyoiku University

Seismograms at Osaka Kyoiku University (on a hill top)

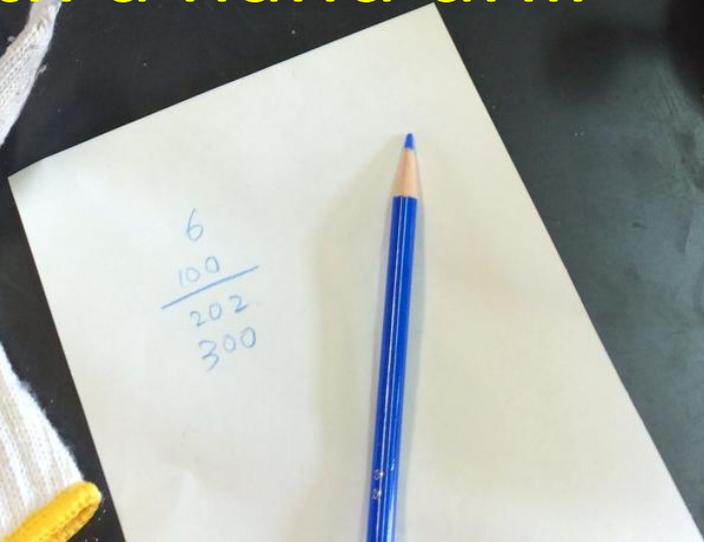
Microseisms and noises are scarce
due to rigid ground condition.

If the air conditioner turns on-----



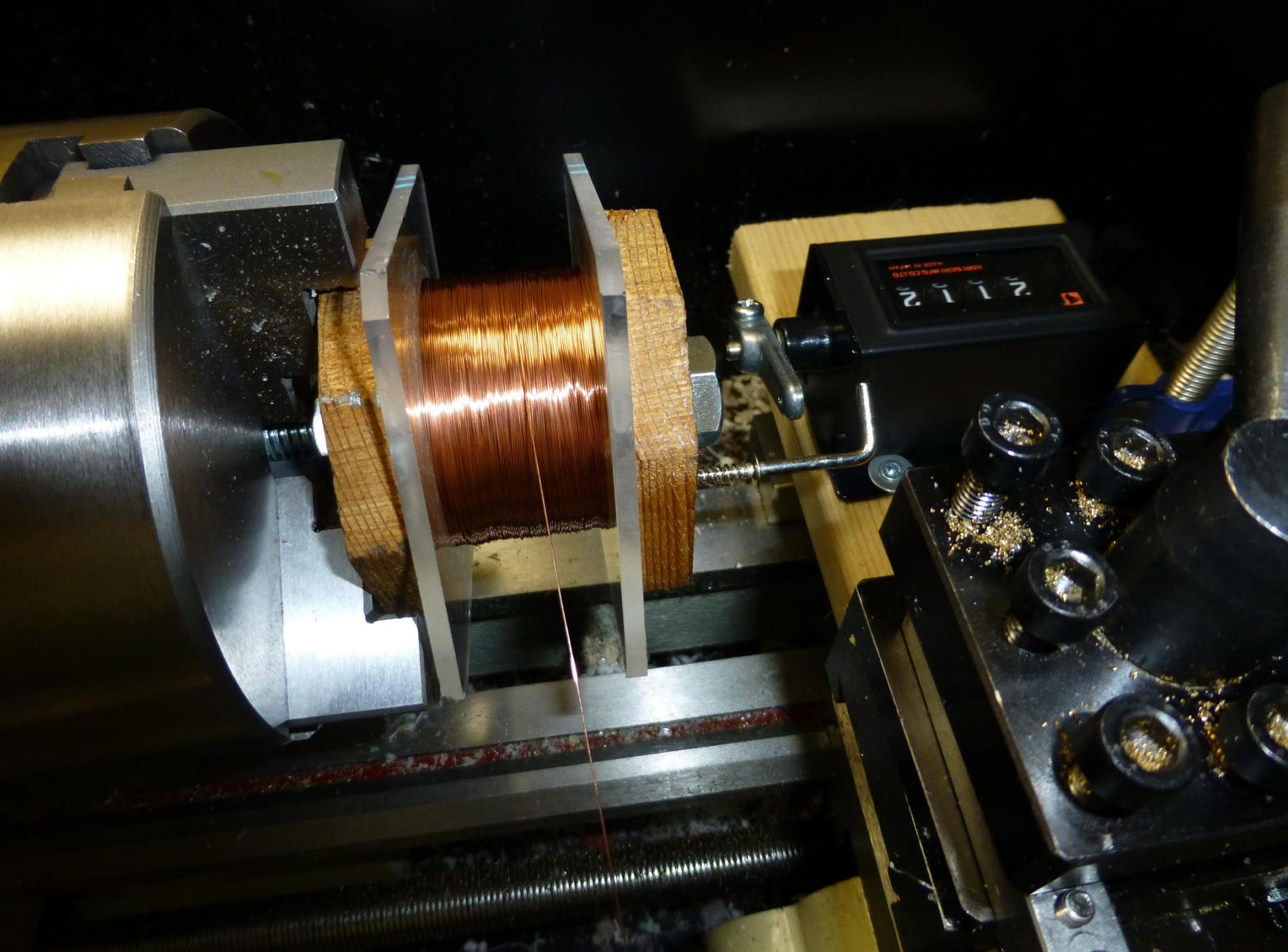
Making a coil with a hand drill

6
100
202
300



Making a coil with a bench lathe

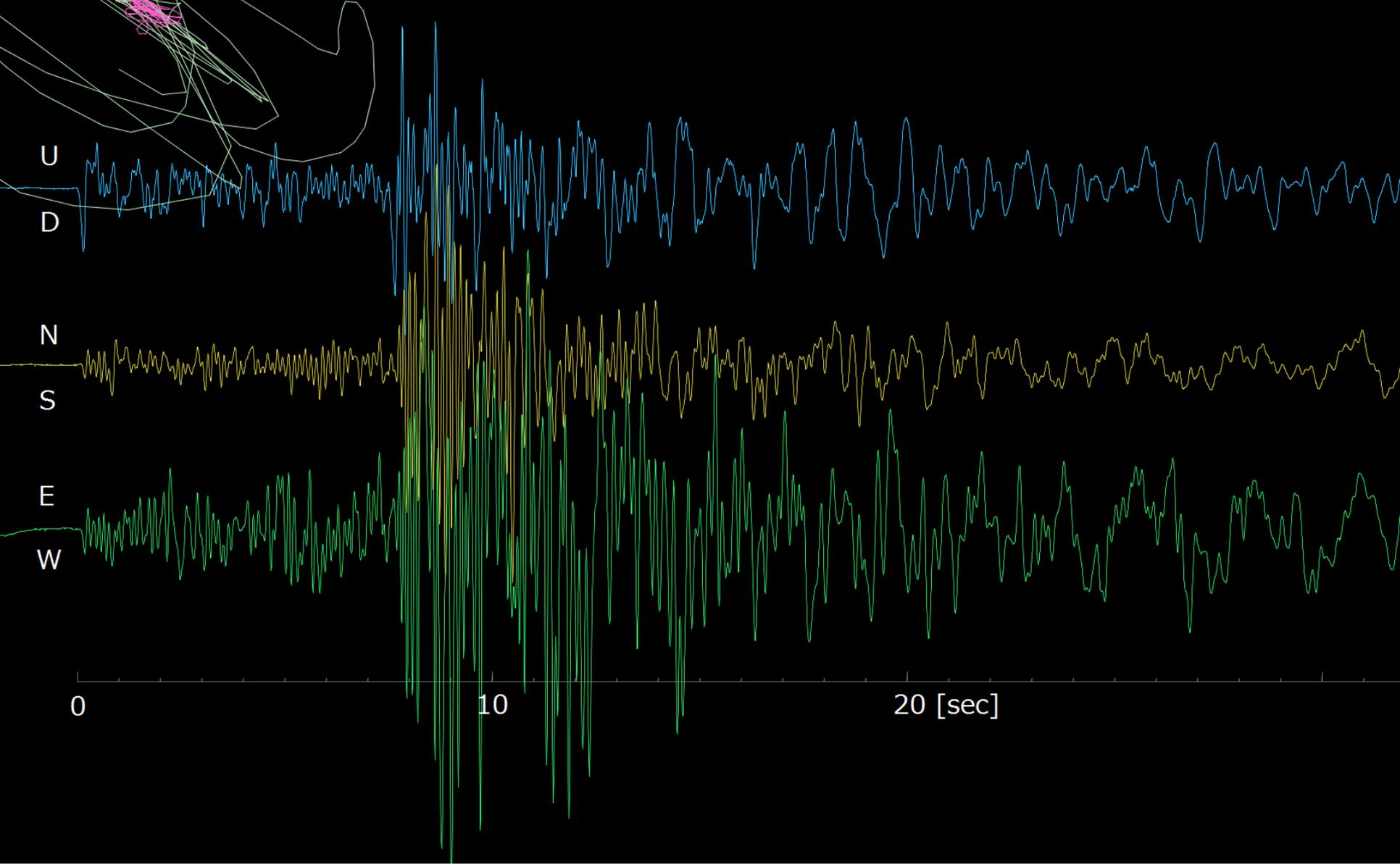




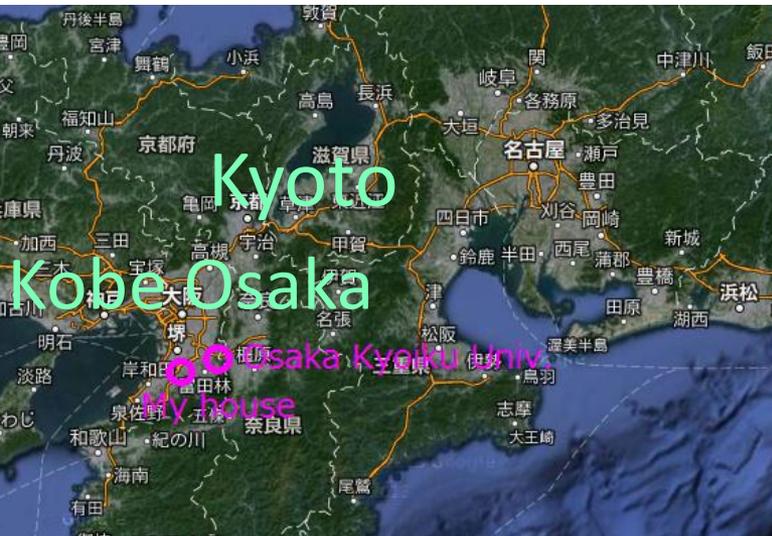
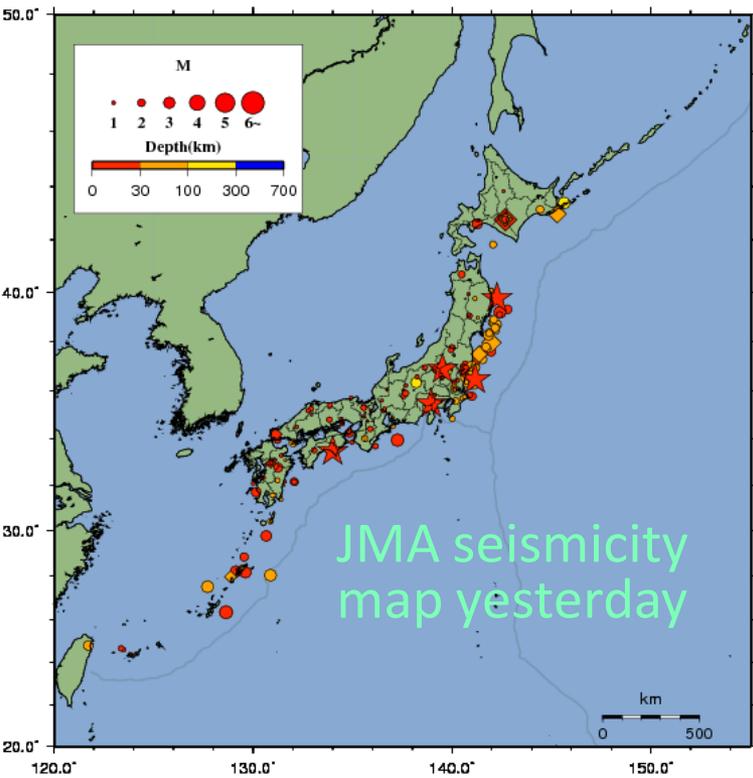
Seismogram examples:

/home/seagull/Desktop/Sayama_select/8/26/Time=03_33_24_IFS

Awajishima reg. Mj 4.1 d=11km $\Delta=63$ km recorded at my house

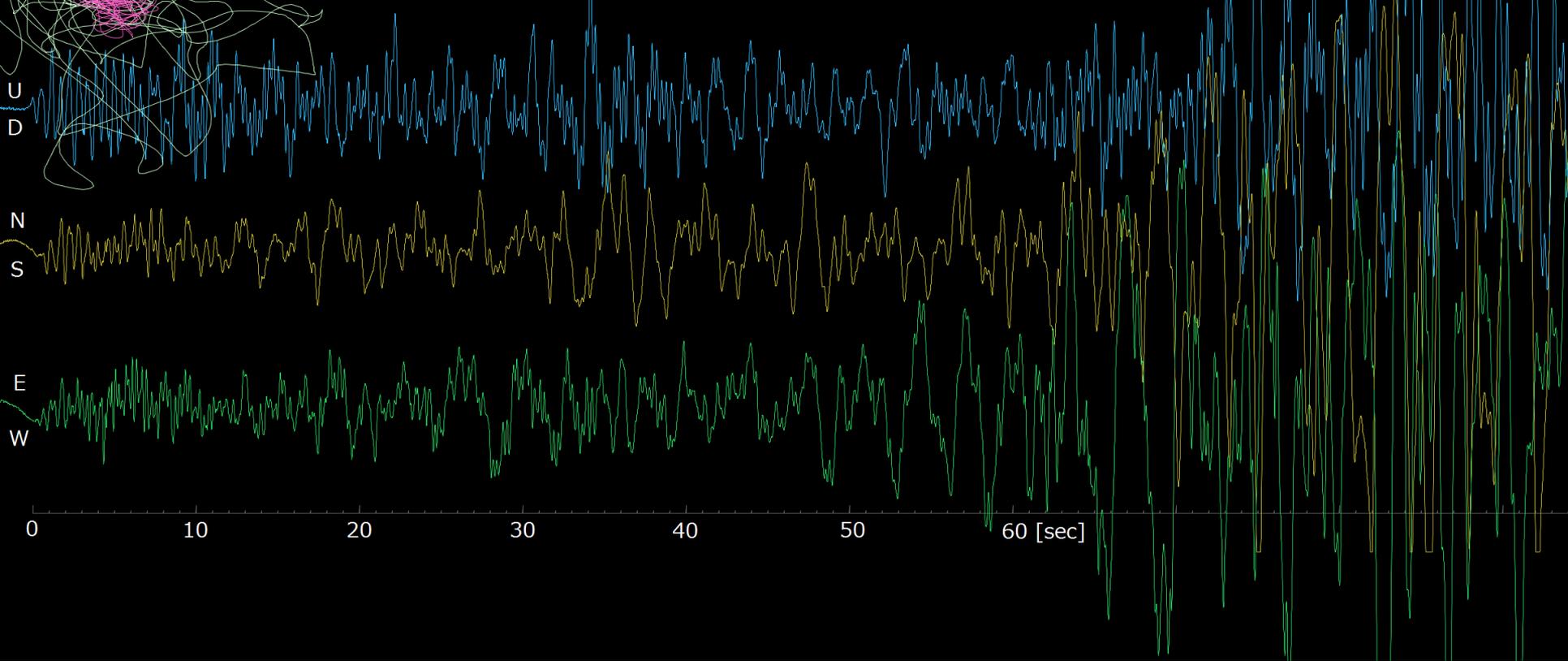


2014/09/04 00:00 ~ 2014/09/05 09:10



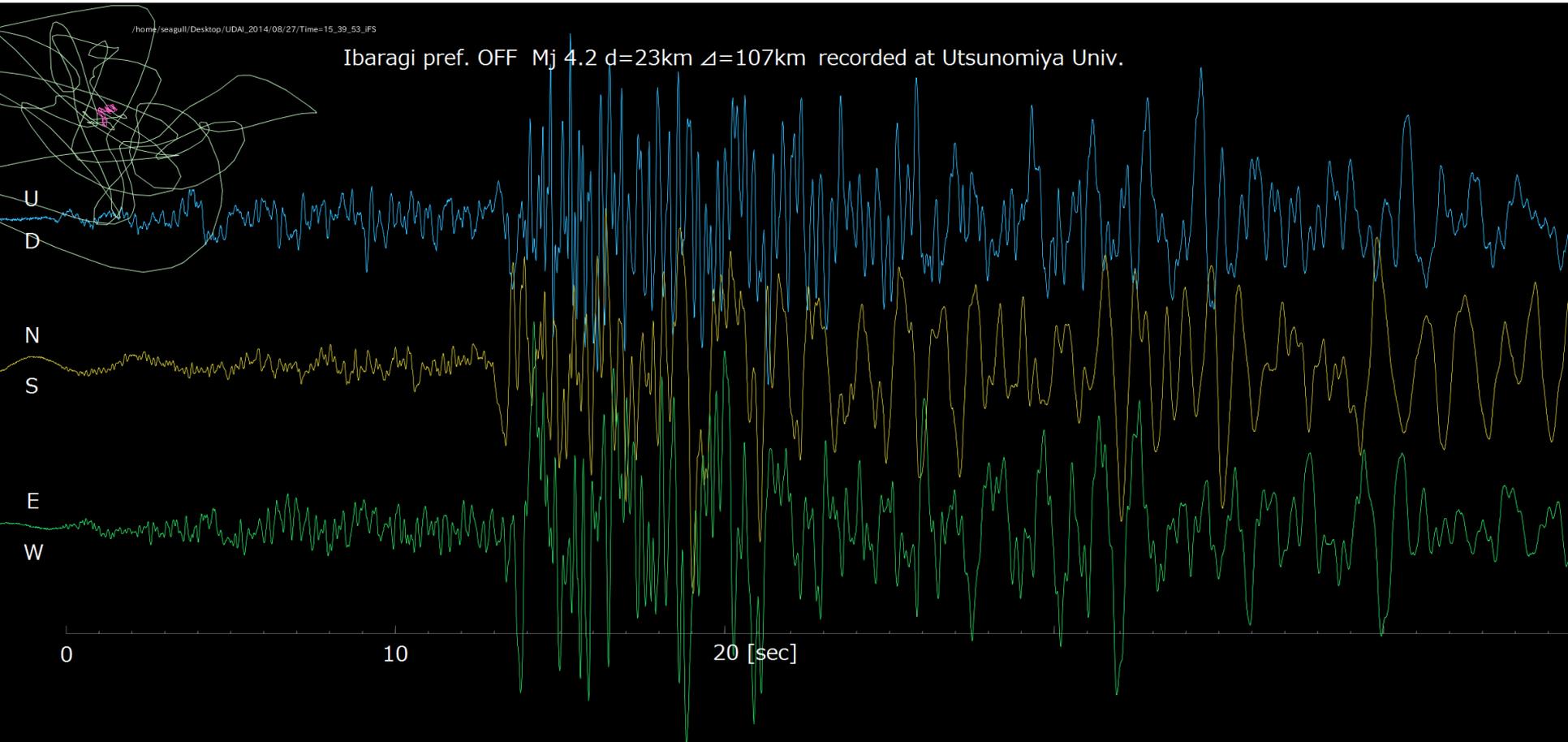
/home/seagull/Desktop/UDAI_2014/08/10/Time=12_41_26_IF5

Aomori fref. E OFF Mj 6.1 d=50km $\Delta=549$ km recorded at Utsunomiya Univ.



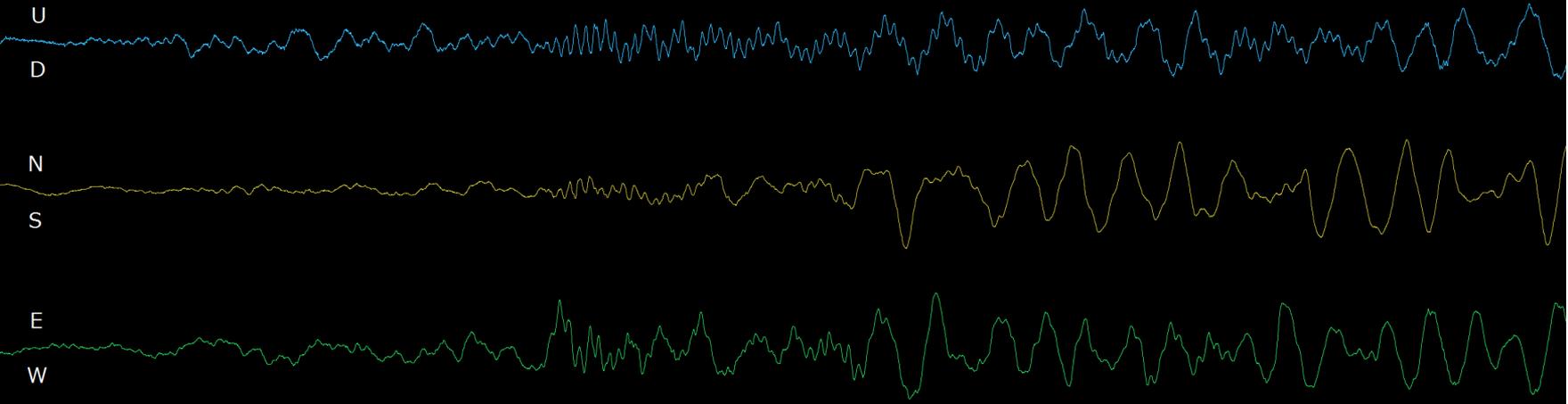
/home/seagull/Desktop/UDAI_2014/08/27/Time=15_39_53_IFS

Ibaragi pref. OFF Mj 4.2 d=23km Δ =107km recorded at Utsunomiya Univ.

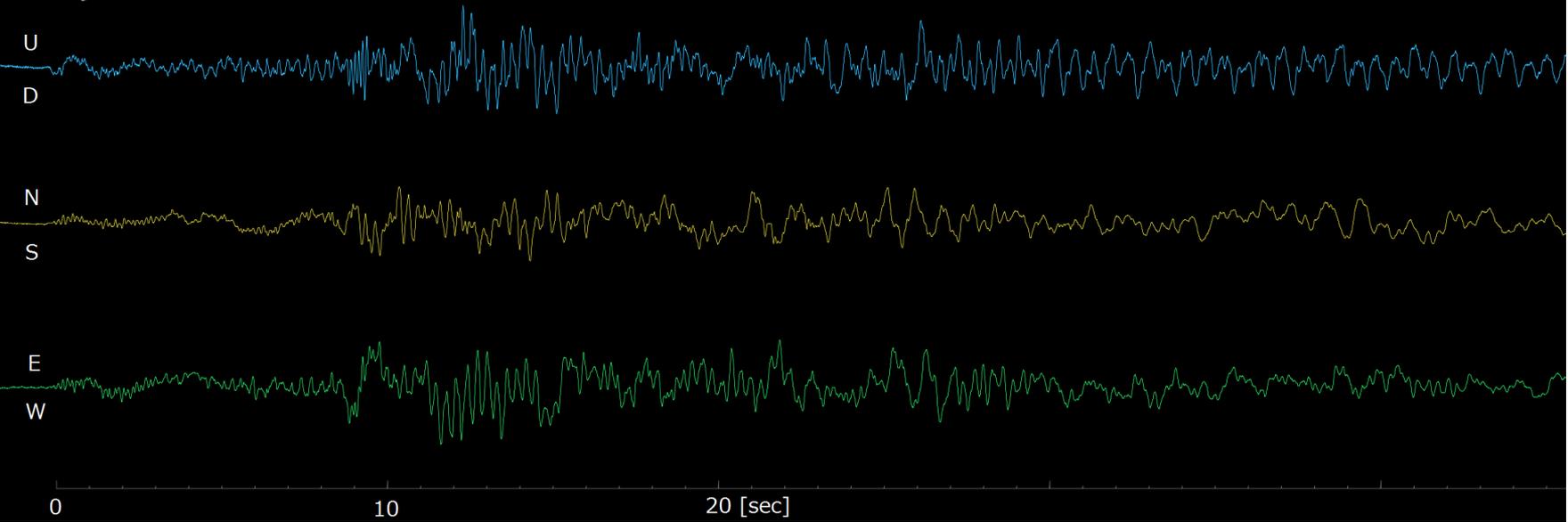




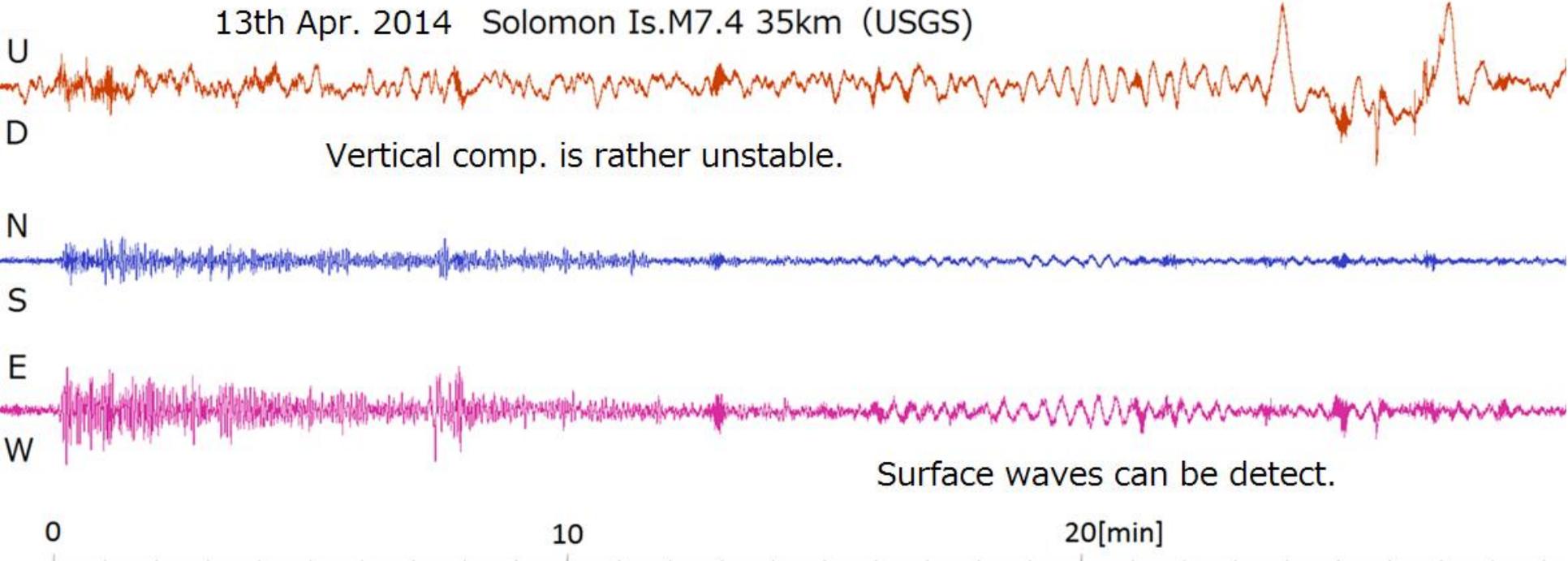
Fukushima OFF Mj 3.7 d=28km Δ=140km recorded at Utsunomiya Univ.



Northern Ibaragi Pref. Mj 3.5 d=55km Δ=61km recorded at Utsunomiya Univ.



An example of foreign earthquakes



Performance

The performance of our system can record **M5.5 or greater of domestic earthquakes**, **M7.5 or greater of foreign earthquakes**, respectively.

Our system particularly horizontal pendulum makes quite stable observation, however the vertical sensor(now construction) and the signal process have some problems about stability.

Discussion

- Our pendulum is quite **fragile against rotational ground motion**, also has **parasitic oscillations in principle**. => We assume that the ground motion has no rotational components.

Because; wave lengths of EQ. ($10^2\text{m} <$) \gg pendulum size ($\sim 10^0\text{m}$).

=> Of course a test on the shaking table is desirable.

- **Parasitic swings and wind stirs** may be the most nuisance problems for our pendulum. => some improvements are necessary.
- Our system should be improved for more precise apparatus. -> Easy making concept vs. precise observation has an annoying trade off.
- The use of **neodym magnet** is necessary, because its strong magnetic flux ($\sim 400\text{ mT}$) supplies a sufficient induced current and an adequate damping.
- The “Arduino” leaves more ability in A/D resolution and sampling rate.

Conclusions

- We developed a new kind of **simple, low-cost, stable and easy making** horizontal seismograph system.
- Simple structure clearly shows a **principle of seismometer** and also shows a data process of **modern digital seismograph system** for students.
- This system is useful for both **crass room demonstration** and daily **observation of natural earthquakes**.
- The **displacement wave forms** instead of velocity records are more appropriate for educational use.
- The wave data recorded by our system will be distributed as crass-room exercises or outreach resources.
- Some improvements are desired for more reliable observation.
- **“Arduino” and “processing”** are the best combination for educational tools **not only** for seismology **but also** for meteorology or geophysical related subjects.

<Acknowledgements and References>

- We thank to Dr. Osamu Wakishima of the Osaka Science Institute of Education for his helpful comments and discussions about making our magnetic circuit.
- This study is supported by Kakenhi JAPAN No.25350200.
- Yoshio OKAMOTO: 「Let's make a seismometer with a film case!」No.1-4, *Nawifuru, a bi-monthly letter for public*, Japan Seismological Society, Vol.0, 5, 1997 (in Japanese) .
- Yoshio OKAMOTO: abstract for GeoSciEdu3 in Sydney, 2000
<http://www.cc.osaka-kyoiku.ac.jp/~yossi/doc/Seismograph.pdf>

Thank for your attention!

- The contents of our study will be available at our web site soon:

<http://www.osaka-kyoiku.ac.jp/~yossi/>

- Any comment or Mail to:

yossi@cc.osaka-kyoiku.ac.jp