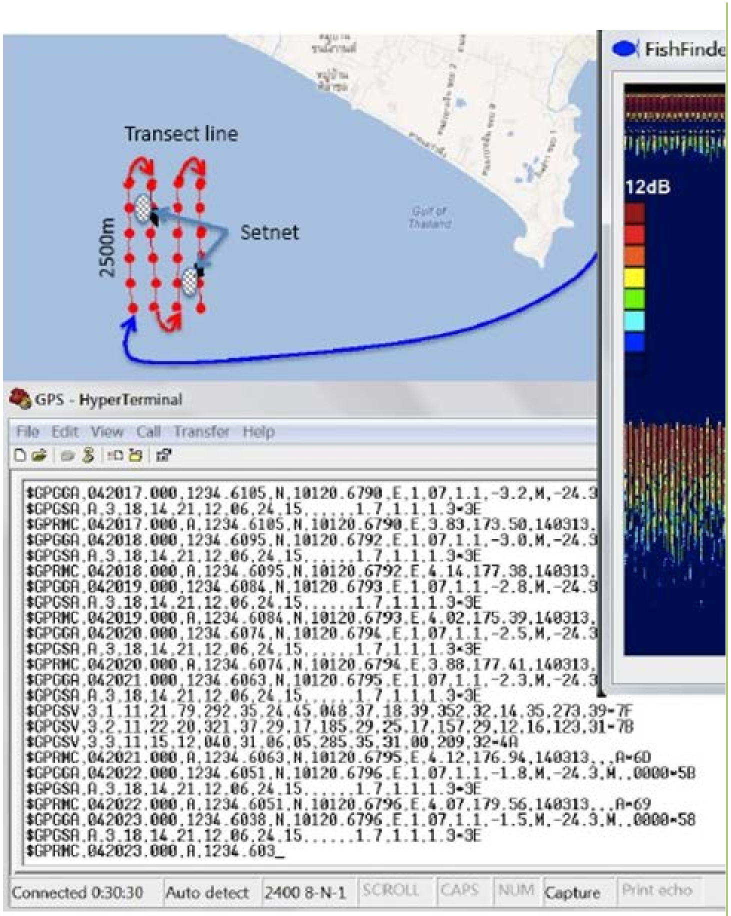
GUIDE TO OPERATION OF ACOUSTIC DATA COLLECTION SYSTEM (AQFI-1301) FOR SHALLOW WATERS



```
$GPGGA,042017.000,1234.6105,N,10120.6790,E,1,07,1.1,-3.2,M,-24.3
$GPGSA,A,3,18,14,21,12,06,24,15,.....,1.7,1.1,1.3*3E
$GPRMC,042017.000,A,1234.6105,N,10120.6790,E,3.83,173.50,140313,
$GPGGA,042018.000,1234.6095,N,10120.6792,E,1,07,1.1,-3.0,M,-24.3
$GPGSA,A,3,18,14,21,12,06,24,15,.....,1.7,1.1,1.3*3E
$GPRMC,042018.000,A,1234.6095,N,10120.6792,E,4.14,177.38,140313,
$GPGGA,042019.000,1234.6084,N,10120.6793,E,1,07,1.1,-2.8,M,-24.3
$GPGSA,A,3,18,14,21,12,06,24,15,.....,1.7,1.1,1.3*3E
$GPRMC,042019.000,A,1234.6084,N,10120.6793,E,4.02,175.39,140313,
$GPGGA,042020.000,1234.6074,N,10120.6794,E,1,07,1.1,-2.5,M,-24.3
$GPGSA,A,3,18,14,21,12,06,24,15,.....,1.7,1.1,1.3*3E
$GPRMC,042020.000,A,1234.6074,N,10120.6794,E,3.88,177.41,140313,
$GPGGA,042021.000,1234.6063,N,10120.6795,E,1,07,1.1,-2.3,M,-24.3
$GPGSA,A,3,18,14,21,12,06,24,15,.....,1.7,1.1,1.3*3E
$GPGSV,3,1,11,21,79,292,35,24,45,048,37,18,39,352,32,14,35,273,39*7F
$GPGSV,3,2,11,22,20,321,37,29,17,185,29,25,17,157,29,12,16,123,31*7B
$GPGSV,3,3,11,15,12,040,31,06,05,285,35,31,00,209,32*4A
$GPRMC,042021.000,A,1234.6063,N,10120.6795,E,4.12,176.94,140313,,,A*6D
$GPGGA,042022.000,1234.6051,N,10120.6796,E,1,07,1.1,-1.8,M,-24.3,M,,0000*5B
$GPGSA,A,3,18,14,21,12,06,24,15,.....,1.7,1.1,1.3*3E
$GPRMC,042022.000,A,1234.6051,N,10120.6796,E,4.07,179.56,140313,,,A*69
$GPGGA,042023.000,1234.6038,N,10120.6796,E,1,07,1.1,-1.5,M,-24.3,M,,0000*58
$GPGSA,A,3,18,14,21,12,06,24,15,.....,1.7,1.1,1.3*3E
$GPRMC,042023.000,A,1234.603_
```

Development and improvement of acoustic equipments and systems for shallow areas

Research Institute for Humanity and Nature (RIHN)

GUIDE TO OPERATION OF ACOUSTIC DATA COLLECTION SYSTEM (AQFI-1301) FOR SHALLOW WATERS

Development and improvement of acoustic equipments and systems for shallow areas

Researchers:

Yoshinori Miyamoto[1], Keiichi Uchida[1],
Yuttana Theparoonrat[2], Monton Anongponyoskun[3],
Kritsada Thongsila[4], Yap Minlee[1], Toyoki Sasakura[5]
Kohei Hasegawa[6]

[1]Tokyo University of Marin Science and Technology, Japan
[2] Southeast Asian Fisheries Development Center, TD, Thailand
[3] Faculty of Fisheries, Kasetsart University, Thailand
[4]Eastern Marine Fisheries Research and Development Center, DOF, Thailand
[5]Fusion Inc., Japan
[6] National Research Institute of Fisheries Engineering,
Japan Fisheries Research and Education Agency, Japan

FOREWORD

Marine resources are important protein sources for the growing human population, but these resources can be difficult to manage because of factors such as their inaccessibility and variability. Acoustic survey technology can help to overcome some of these difficulties. In particular, the development of acoustic equipment and analytical systems to process the acoustic data have been very beneficial for marine stock assessments, especially in deep areas such as the high seas. Here, I would like to mention other benefits of applying new acoustic technologies towards natural resource management in coastal areas of tropical zones.

Although coastal areas are relatively small in relation to the area of the open ocean, they contain high amounts of biodiversity and primary productivity and thus are very important for resource sustainability. In general, many small-scale fisheries are conducted in coastal areas, and these fisheries play important roles as income and protein sources for local people. Unfortunately, collecting data on small-scale fisheries can be quite difficult, in part, because much of the activity is family-based, the products are traded during face-to-face interactions. Therefore, collaboration is needed among the local fishers, researchers and governmental officers to collect reliable fisheries statistics data toward sustainable fisheries developments. In this regard, it is important to find better ways to involve the local fishers in stock assessments and fisheries management decisions.

A fishery management typically begins with a stock assessment based on statistical data, and then, regulate fishing activities up to target yields e.g. MSY etc. Authorities responsible for fisheries management then enforce the corresponding regulations. This approach seems quite rational, but it does not do enough to facilitate the participation of local fishers. There are typically no incentives for local fishers to collaborate with the managing authorities. The establishment of a system in which local fishers are given high incentives to participate in resource management would be very beneficial for coastal fisheries.

A community-based set-net fishery was installed along a beach in Rayong, Thailand, in 2003, and catch data from this fishery during 2003 to 2013 were well collected by fishermen. This success shows us how best to involve local fishers in data collection efforts. Government officials and researchers provided some initial technical and economic support for the set-net installation, and local fisheries groups were directly involved in product sales and community-based activities such as data collection. Catch data and selling data were shared among the members. Moreover, the set-net could act as a nursery for larval and juvenile fish, and as a physical barrier against illegal fishing by large vessels in the coastal zone. The members of the set-net fisheries groups demonstrated concern about the fish inside the set-net. Acoustic technology was employed to give them information about the actual number of fish in the net. If the technology actually installed to set-net, incentive of fishermen to collaborate with researchers will also be enhanced because this technology improved the efficiency of fisheries management efforts.

If local fisher groups are able to use acoustic technology to assess and monitor the fishery resources, they are better able to understand the status of the fisheries and they can utilize the resources effectively. As the depth of the coastal area is shallow in many locations, traditional acoustic surveys for stock assessments are not always a suitable option. However, some acoustic survey systems and equipment have been invented recently those are applicable to shallow waters. Even though these shallow-water systems do not have high resolution capabilities like the systems used in high sea areas, they can provide useful information and help communities visualize the fisheries resource situation in the underwater environment. Such information was found to be important for unifying the local fishers who operate in the same fishing grounds.

Recently, area-capability cycle was proposed as a model for balanced development in rural areas, The area-capability cycle consists of finding new resources, effectively using them, developing a community of users, enhancing user capability, cultivating interest in the health of the ecosystem, understanding the importance of caring for the environment, promoting activities to care for natural habitats and primary production of resources, and fostering pride and hope in users in regards to their use and care of ecosystem services. New technology can help find new resource; new technology can facilitate effective utilization of the resource. And new technology can promote collaboration of many stakeholders. What I would like to say is the development of acoustic technology can be a key for new coastal development model.

Satoshi ISHIKAWA

Research Institute for Humanity and Nature (RIHN)

CONTENTS

※ About trademarks: FURUNO, RotoKey are trademarks or registered trademarks of Furuno Electric Co., Ltd.

INTRODUCTION

Research Institute for Humanity and Nature (RIHN) program,
Component 5 Activities:

Activity1: Developing the new data collection equipments and analysis systems of acoustic survey at coastal areas
Activity2: Field test of the developed equipments and system
Activity3: On-site-Training on new acoustic survey equipments and systems
Activity4: Publication of the research protocol guidelines for acoustic survey in coastal areas

Development and improvement of acoustic equipments and systems for shallow area

This study was carried out as part of the RIHN project on “Coastal area-capability enhancement in Southeast Asia”. This project was aimed at investigating the linkage between livelihoods and ecosystem health in the Southeast Asian coastal areas to fully understand its complexity and consequent vulnerability, particularly from the human-related viewpoints through collaborative holistic researches with local peoples. The acoustic data collection system including equipment modification was conducted for coastal area survey around the set-net fishing ground in Rayong Province, Thailand. An analytical methodology was also developed and used as a tool for acoustic survey methodology and education for young scientists. Since, the depth of the target research area is about 15 meters, the searching range is too narrow for using scientific echo sounder available in the market, the project therefore modified the acoustic device which is composed of GPS Plotter Fish-Finder (FURUNO GP1670F), Interface box and personal computer (PC), in order that this could be used as acoustic data collection equipment in coastal areas.

Objectives

1) To develop new acoustic data collection system for shallow waters,
2) To study the fisheries resources distribution around set-net fishing ground in Rayong Province, Thailand,
3) To estimate the amount of fisheries resources using acoustic data and the fish catch data of set-net fishing operation, and
4) To conduct human resource development activities on the new acoustic survey equipment and system through on-site training and publication of guidelines for acoustic survey in coastal areas.

ACOUSTIC DATA COLLECTION SYSTEM FOR SHALLOW WATERS

Development of new data collection equipments and analysis systems for acoustic survey in coastal areas.

Development of acoustic data collection system and modification of equipment were carried out for the coastal area survey in Rayong Province, Thailand. The hardware and software system for data collection were developed at Tokyo University of Marine Science and Technology, Tokyo Japan. The first testing session of the hydro-acoustic equipments and systems for shallow areas were conducted at Tateyama Bay, Chiba Prefectures, Japan in October 2012.

Modification of GPS and echo sounder system for hydro-acoustic data collection

The hydro-acoustic system for data collection was modified by using any kind of echo sounder and GPS receiver. Some devices have echo sounder and GPS plotter (e.g. FURUNO GP-1670F), but some devices are independent of an echo sounder and the GPS receiver.

Required specifications of echo sounder

The echo sounder should operate on 50 kHz. The data collection system was designed to record the echo sounder signal of 50 kHz only.

Required specifications of GPS receiver

The GPS receiver provides highly accurate position, courses and speed information. The GPS receiver should have these data output. Normally, these data are in NMEA0183 format and out -put in RS-232C or USB terminal are used for other instruments.

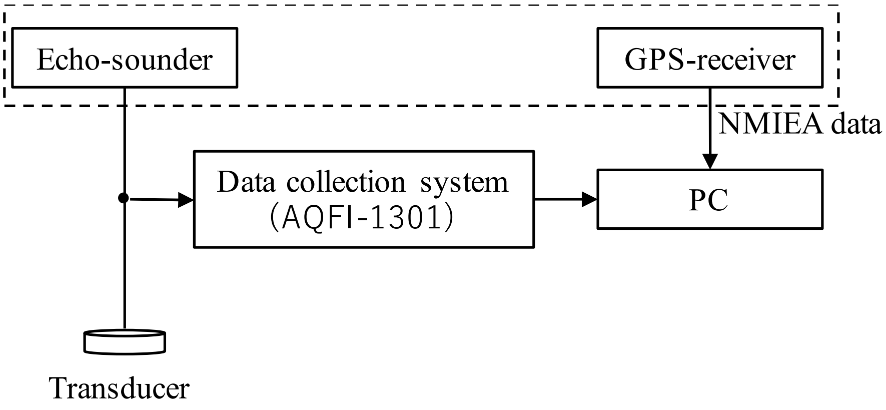


Figure 1. Echo sounder data collection system block diagram.

In this study, the hydro-acoustic system for data collection was modified by using FURUNO GPS Plotter model GP-1670F. The GP-1670F is equipped with GPS receiver and chart plotter system. The machine is also equipped with echo sounder operating simultaneously on 50 kHz and 200 kHz. The GP-1670F provides a total integrated GPS receiver, color video plotter and color fish finder. The built-in GPS receiver provides highly accurate position, courses and speed information. The fish finder presents vivid underwater images on a high quality LCD.

The compact display unit and antenna unit permit could be installed even where space is limited. Specifications of the equipment are shown as follows,

Main features of GP-1670F;

- Bright 5.7 inch (GP-1670F) color LCD with brilliance control.
- Excellent viewing angles, even when wearing sunglasses.
- Internal GPS receiver provides high accurate position information (GPS, within 2.5 m, SBAS, within 2 m).
- Customizable analog and digital displays show wind angle and speed, engine condition (speed, temperature, oil pressure, etc.), etc.
- Large internal memory stores 30,000 track points, 30,000 points, 1,000 routes (500 waypoints/route).
- SD card slot accepts SD and SDHC cards for external storage of data and settings.
- Full range of alarms; Arrival, Anchor Watch, Cross-track Error, Speed, Depth, Temperature, Fish Alarm, Bottom Alarm, etc.
- Man overboard (MOB) feature records latitude and longitude coordinates at the time of MOB.
- CAN bus interface for the connection of GPS Receiver, Weather Station, FI-50 (instrument series), Satellite Compass, etc.
- Accepts NMEA0183 input with optional NMEA data converter.
- Internal GPS antenna available.
- C-Map 4Dcharts available.

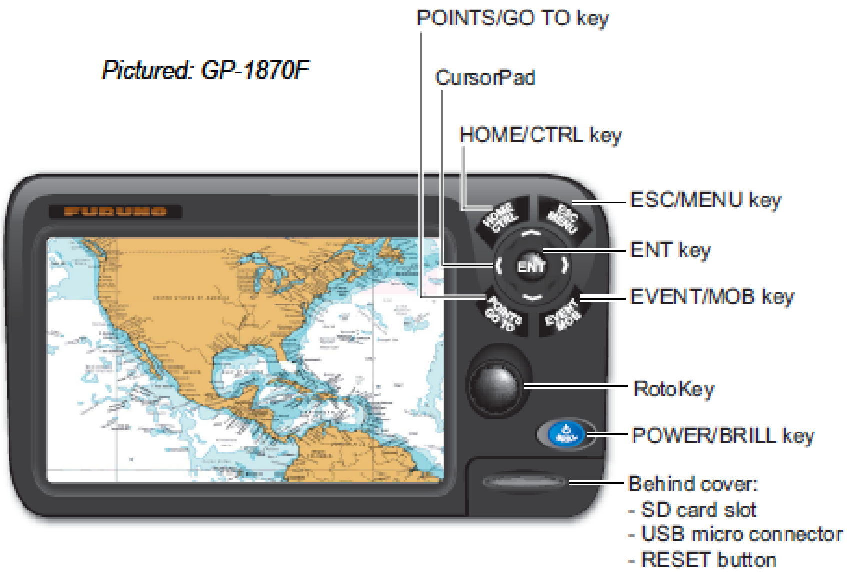


Figure 2. FURUNO GP-1670F display and control panel.

※ This figure is quoted from “FURUNO GPS PLOTTER/SOUNDER GP1670F, GP1870F OPERTOR’S MANUAL” of product GP-1670F of Furuno Electric Co., Ltd.

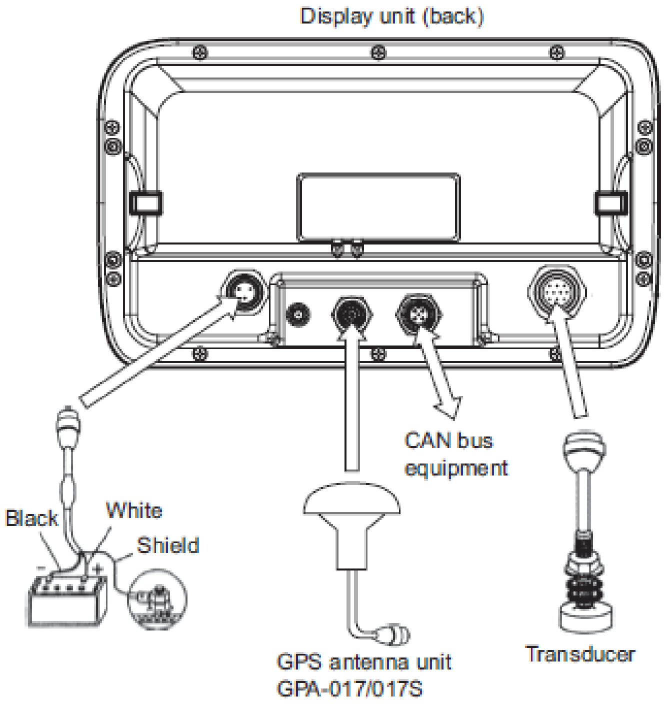


Figure 3. Connection socket of GP 1670F.

※ This figure is quoted from “FURUNO GPS PLOTTER/SOUNDER GP1670F, GP1870F OPERTOR’S MANUAL” of product GP-1670F of Furuno Electric Co., Ltd.

HOW TO CONNECT ECHO SOUNDER TO DATA COLLECTION SYSTEM

There are multiple signal lines on the cable connecting the main unit and the transducer. Among them, it is necessary to find the signal line used for transmission/reception. The number of pins and assignments varies from manufacturer to manufacturer. In many cases, it is described in the manual of each device. After checking the signal line, cut the cable halfway and connect it to terminal 8 (Transducer side) shown in Figure 4. Other signal lines are connected to the same pin number as Transducer side and GP1670F side. If necessary, replace the connector both of Transducer side and GP1670F XDR PORT.

Equipment modification was performed by importing the echo sounder transmitting and receiving signal from transducer cable to the newly designed Echo Sounder Data Collection System (AQFI-1301). The Echo Sounder Data Collection System (AQFI-1301) is composed of Pre-Amplifier and Band Pass Filter, Interface unit, Analog to Digital Converter, and PC computer system. The GPS position data is transferred from GP 1670F to PC Computer passing through NMEA to USB port. The GPS position data are recorded to the PC using Windows Hyper-terminal program. The data collection system is designed to record the echo sounder signal of 50 kHz. only.

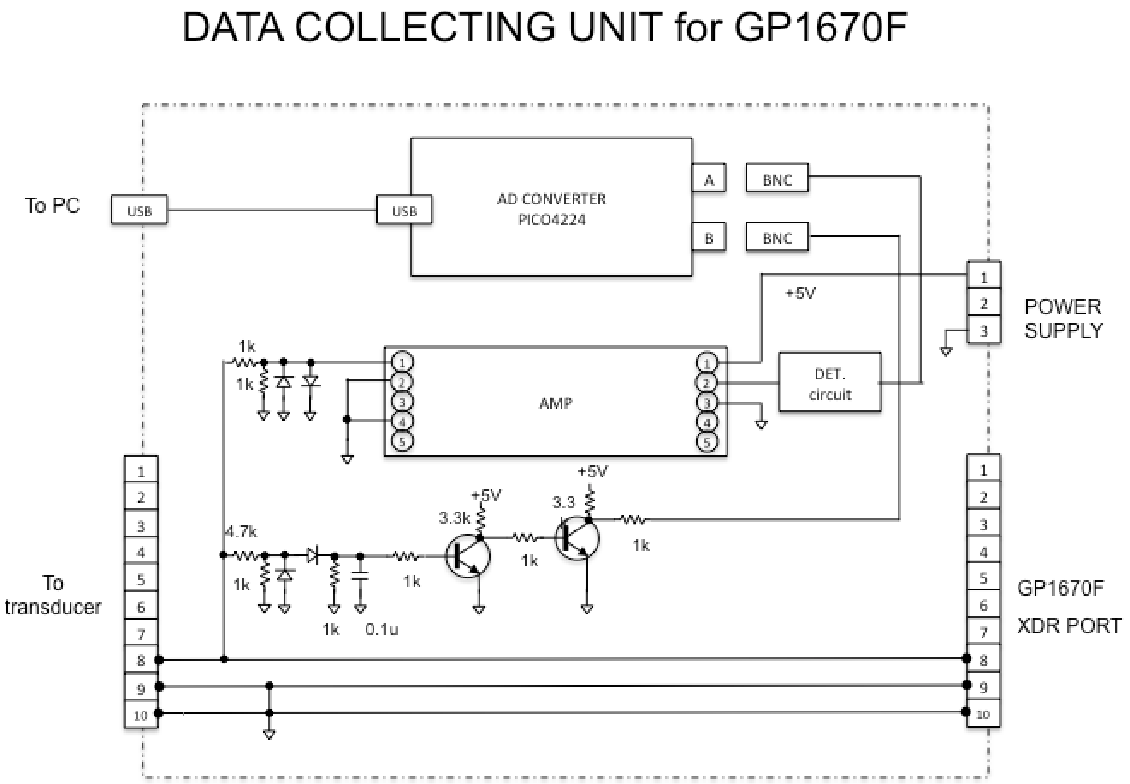


Figure 4. Circuit diagram of Survey Echo Sounder Data Collection System (AQFI-1301).

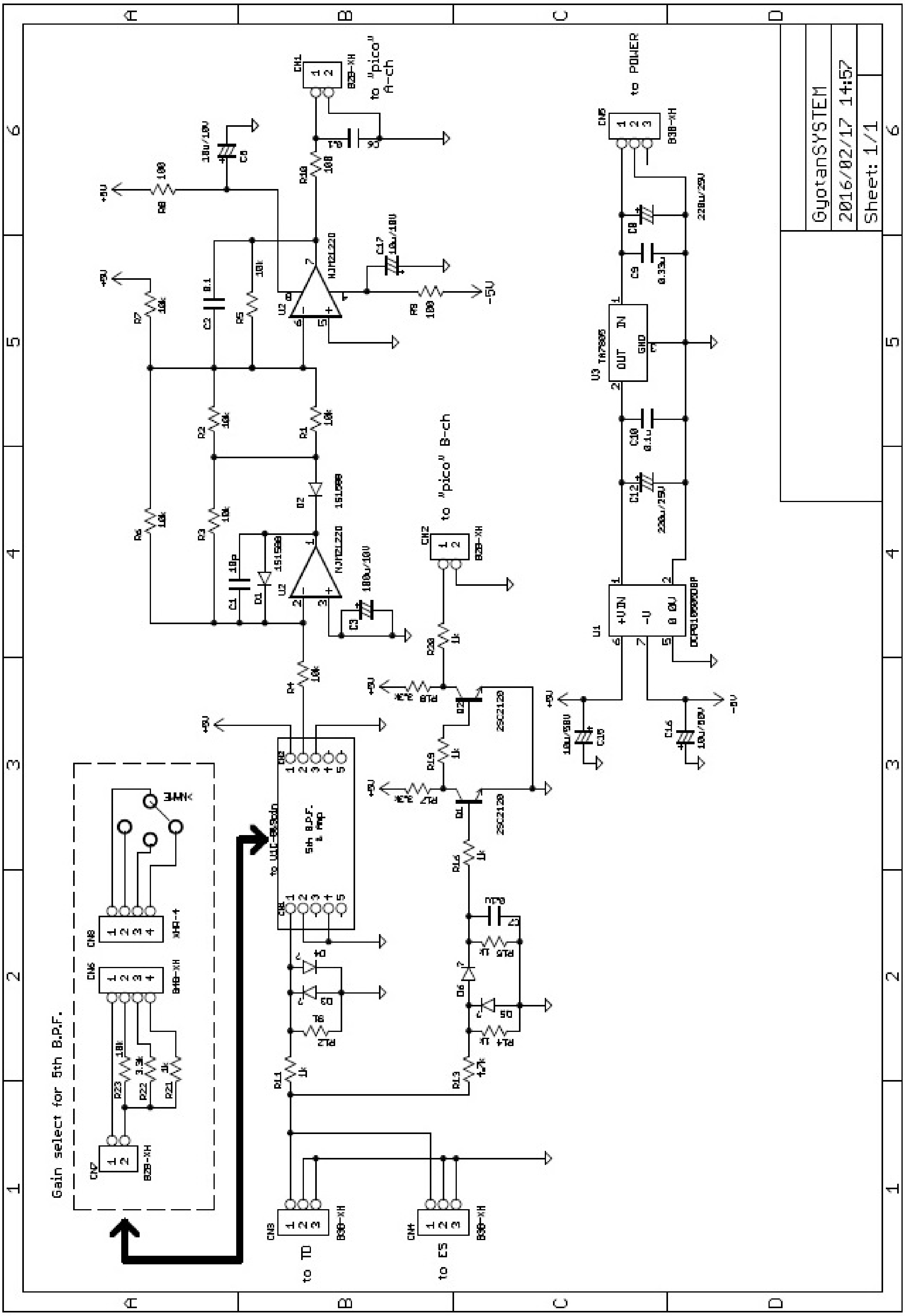
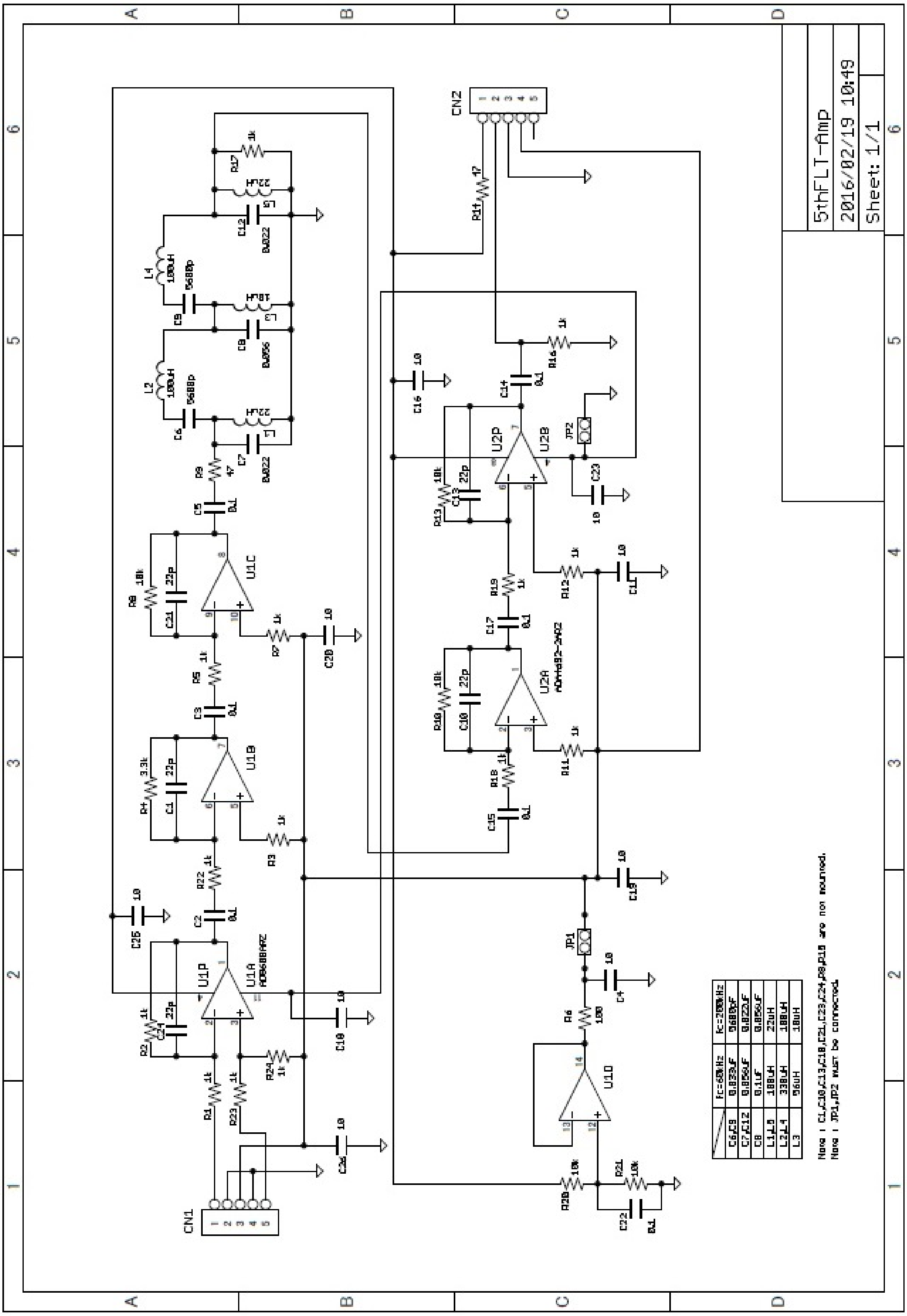


Figure 5. Frequency filter circuit at 50 kHz**.**



	Fc=60kHz	Fc=200kHz
C6,C9	0.033uF	5600pF
C7,C12	0.056uF	0.022uF
C8	0.1uF	0.056uF
L1,L5	100uH	22uH
L2,L4	330uH	100uH
L3	56uH	10uH

Note : C1,C10,C13,C18,C21,C23,C24,R8,R15 are not mounted.
Note : JP1,JP2 must be connected.

Figure 6. Pre-Amplifier circuit at 50 kHz**.**

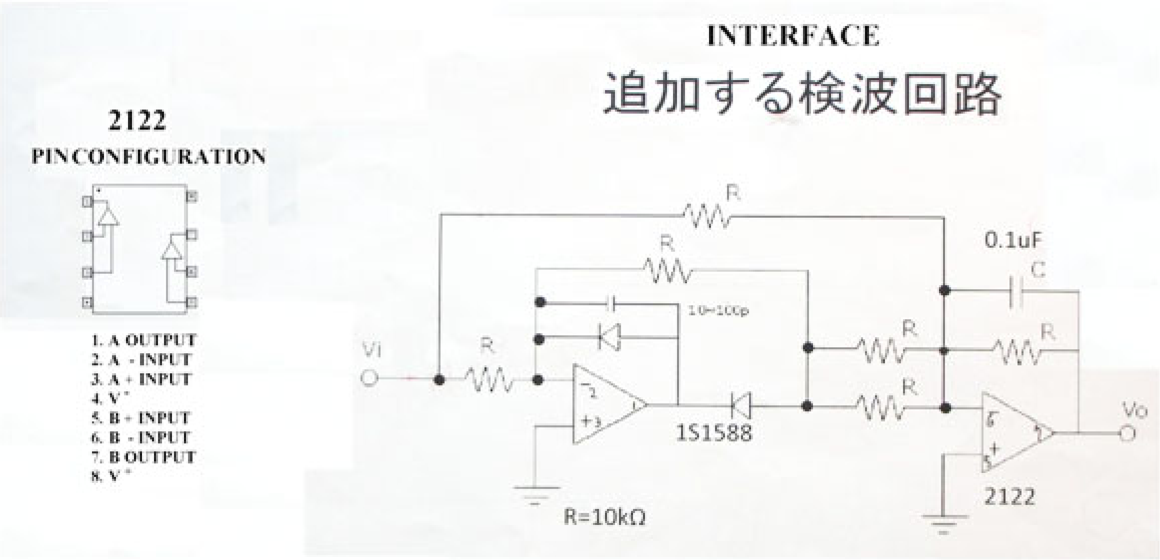


Figure 7. Interface circuit using IC 2122**.**

The return echo signal picks-up from the transducer cable is fed into Pre-Amplifier circuit for signal filtering and amplifying. Analog echo signal from Pre-Amplifier are put into Analog – Digital Converter, Pico 4224, for transforming into echo signal digital data. Personal Computer receives and records digital echo signal by AQUA SOUND software program of "FishFinder, Version 3.2" for Microsoft Windows 7. Digital data from echo sounder are sent to PC through USB port.

.

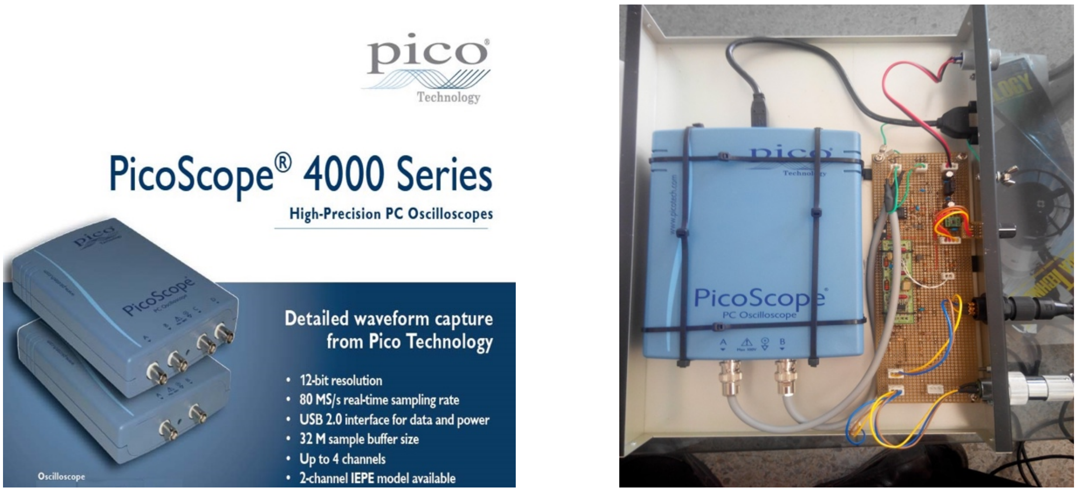

pico
Technology
PicoScope® 4000 Series
High-Precision PC Oscilloscopes
Detailed waveform capture from Pico Technology
• 12-bit resolution
• 80 MS/s real-time sampling rate
• USB 2.0 interface for data and power
• 32 M sample buffer size
• Up to 4 channels
• 2-channel IEPE model available
Oscilloscope
PicoScope
PC Oscilloscope

Figure 8. Analog to Digital Converter, PicoScope 4224.

How to connect GPS receiver to PC

Ship position information (GPS) from echo sounder (GP1670F) is present on the CAN bus equipment port. CAN bus is a communication protocol that shares multiple data and signals through a single backbone cable. By simply connecting any CAN bus devices into the backbone cable, the network onboard can be expanded. All CAN bus devices can be incorporated into the NMEA2000 network.

The GPS receiver should have USB or RS232C data output port. If it is a USB output, connect it directly to the PC. If it is RS232C, connect it to the PC using the conversion cable (Figure 10).

The GPS information from echo sounder is transferred to the PC by applying the program “HyperTerminal” through NMEA Data Converter (IF-NMEA2K2), USB-to-Serial (RS232) Converter, and PC USB port.

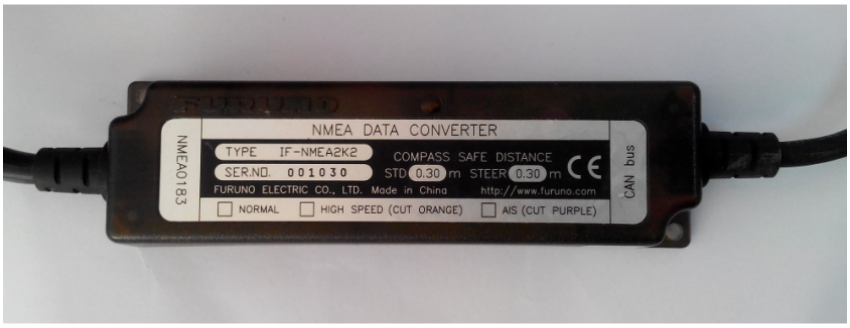


Figure 9. NMEA Data Converter (IF-NMEA2K2).

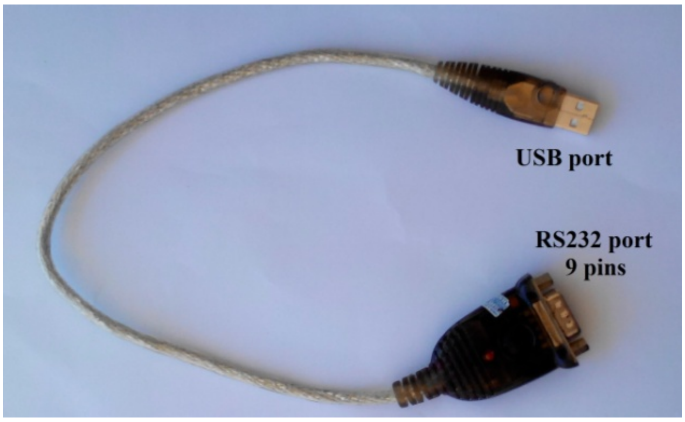


Figure 10. USB-to-Serial (RS232) Converter.

SETTING UP OF ACOUSTIC DATA COLLECTION SYSTEM

The acoustic data collection system hardware comprises of three main components such as Echo sounder (GP-1670F), Data Collection System (AQFI-1301), and Data Collecting Unit (PC). The set up of the actual acoustic data collection system hardware is shown in Figure 11.

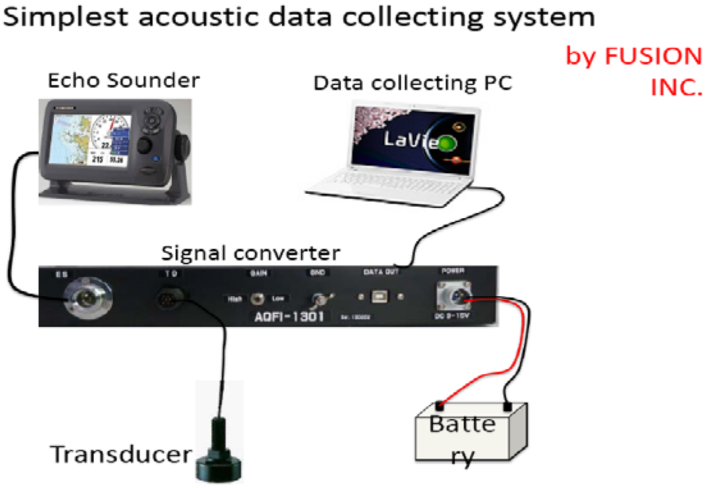


Figure 11. Acoustic data collection system hardware connection layout**.**

Setting up the acoustic data collection system hardware could be carried out following the procedures below:

1. **Power cable connection**
   Connect the power cable to the power connector of FURUNO GP-1670F and Echo Sounder Data Collection System (AQFI-1301). Connect the leads to the battery (12 or 24 VDC): red to plus (+) terminal and black to minus (-) terminal. Make sure that the shield is safe by connecting it to a ground which is the ship's ground.
   Note: Since the fuse (3A) is not waterproof, wrap the fuse holder with vinyl tape to keep water out from the fuse holder.

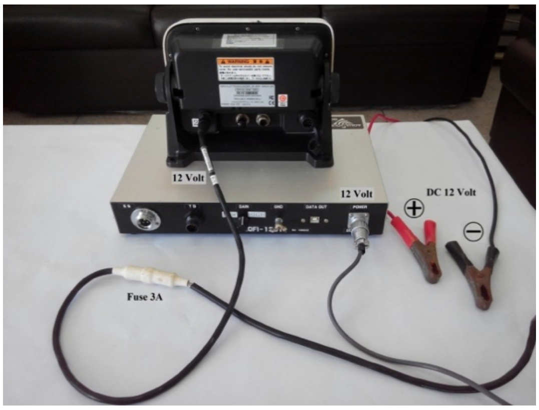


Figure 12. Power cable connection.

2. **Transducer connection**
   Connect the transducer cable to the XDR port of GP-1670F and ES port of AQFI-1301. Connect transducer cable, (520-5PSD) to TD port of AQFI-1301.

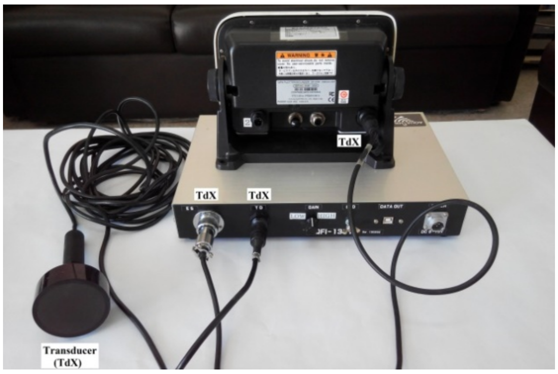


Figure 13. Transducer connection.

3. **CAN bus (GPS Information) connection**
   Connect FURUNO CAN bus cable of NMEA Data Converter (IF-NMEA2K2) to CAN bus port of GP-1670F. Connect RS232 Serial port (9-pins) of NMEA Data Converter (IF-NMEA2K2) to USB-to-Serial (RS232) Converter. Connect USB of USB-to-Serial (RS232) Converter to computer.

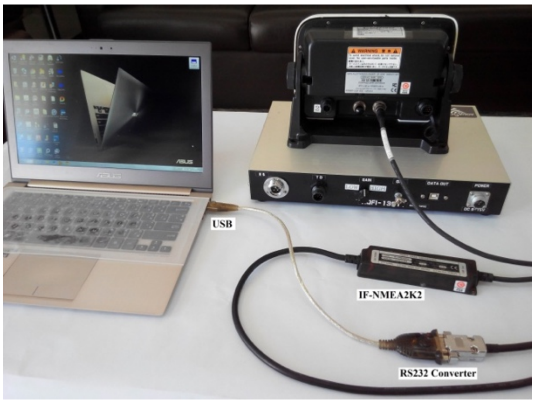


Figure 14. CAN bus (GPS Information) connection.

4. **Data output connection**
   Connect data cable USB type B male to Data Out of Echo Sounder Data Collection System (AQFI-1301) and USB type male A to Computer USB port.

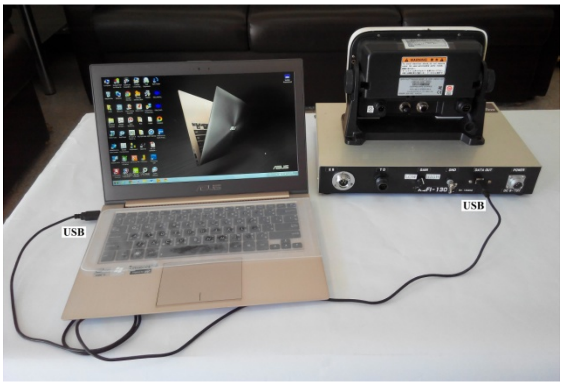


Figure 15. Data output connection.

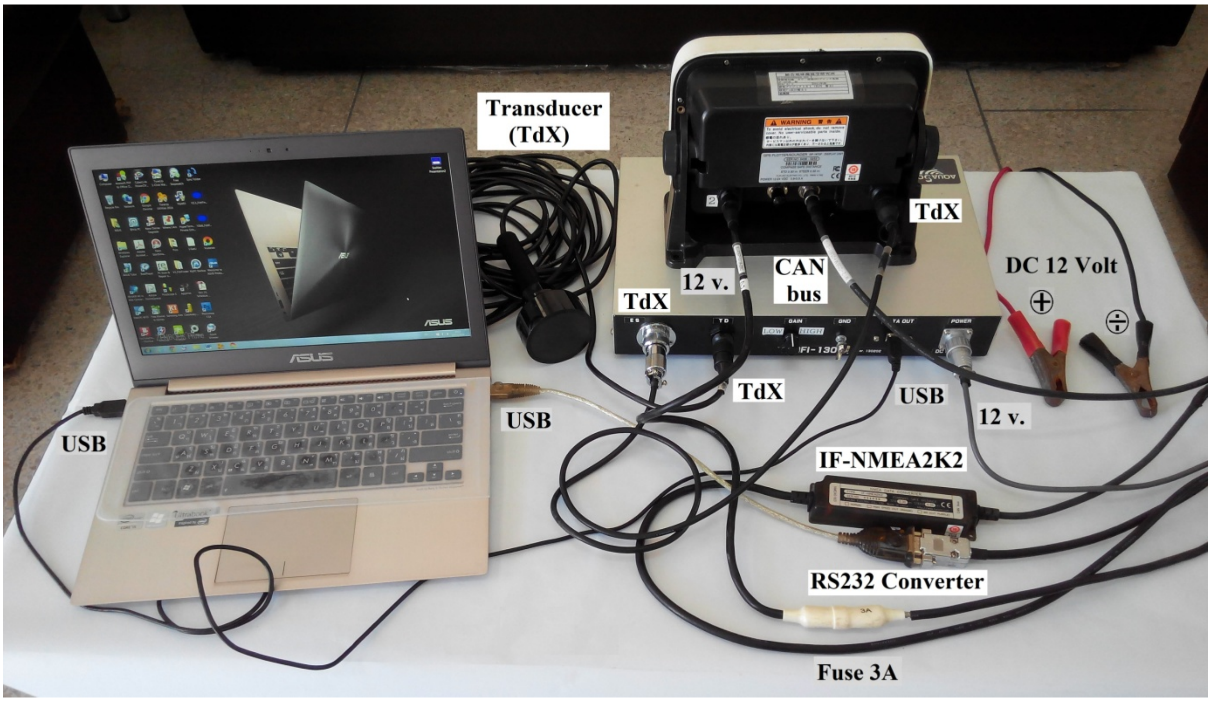


Figure 16. Echo sounder survey data collection system hardware connection.

Mount a Transducer

The thru-hull mount transducer provides the best performance of all, since the transducer protrudes from the hull and the effect of air bubbles and turbulence near the hull skin is reduced. If the boat has a keel, the transducer should be at least 30 cm away from it.

The performance of fish finder is directly related to the mounting location of the transducer, especially for high-speed cruising. Thus, the installation should be planned in advance, keeping in mind the length of the transducer cable and the following factors:

- Air bubbles and turbulence caused by movement of the boat seriously degrade the sounding capability of the transducer. The transducer should, therefore, be located in a position where water flow is the smoothest. Noise from the propellers also adversely affects performance and the transducer should not be mounted nearby. The lifting strakes are notorious for creating acoustic noise, and these must be avoided by keeping the transducer inboard.
- The transducer must always remain submerged, even when the boat is rolling, pitching or planning (gliding) across water at high speed.
- A practical choice would be to place the transducer somewhere between 1/3 and ½ of the boat's length from the stern. In the hull, it is generally practical to install or locate the transducer rather far from the boat stern, so that is always in water regardless of the planning attitude of the boat.

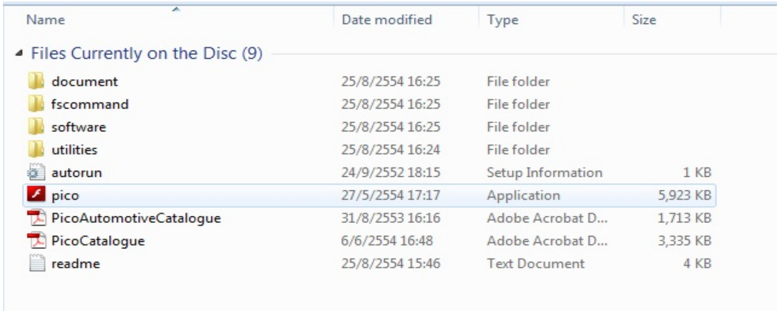
SYSTEM SOFTWARE INSTALLATION AND CONFIGURATION

A. Pico Scope 6

1) Insert Pico Technology system disk to computer

pico
Technology
Product specifications • User manuals • Application notes
Software and Reference CD-ROM
R.50
www.picotech.com

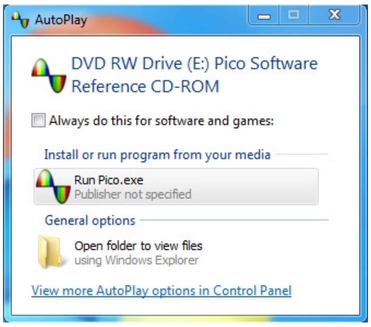
2) Select “pico” Application, Enter



Files Currently on the Disc (9)

Name	Date modified	Type	Size
document	25/8/2554 16:25	File folder	
fscommand	25/8/2554 16:25	File folder	
software	25/8/2554 16:25	File folder	
utilities	25/8/2554 16:24	File folder	
autorun	24/9/2552 18:15	Setup Information	1 KB
pico	27/5/2554 17:17	Application	5,923 KB
PicoAutomotiveCatalogue	31/8/2553 16:16	Adobe Acrobat D...	1,713 KB
PicoCatalogue	6/6/2554 16:48	Adobe Acrobat D...	3,335 KB
readme	25/8/2554 15:46	Text Document	4 KB

3) Select “Run Pico.exe” from AutoPlay, Enter



AutoPlay

DVD RW Drive (E:) Pico Software Reference CD-ROM

□ Always do this for software and games:

Install or run program from your media

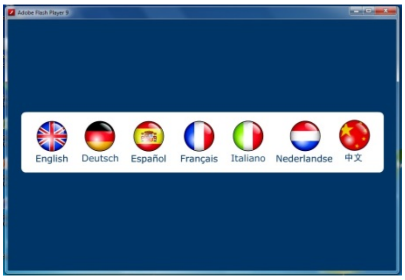
Run Pico.exe
Publisher not specified

General options

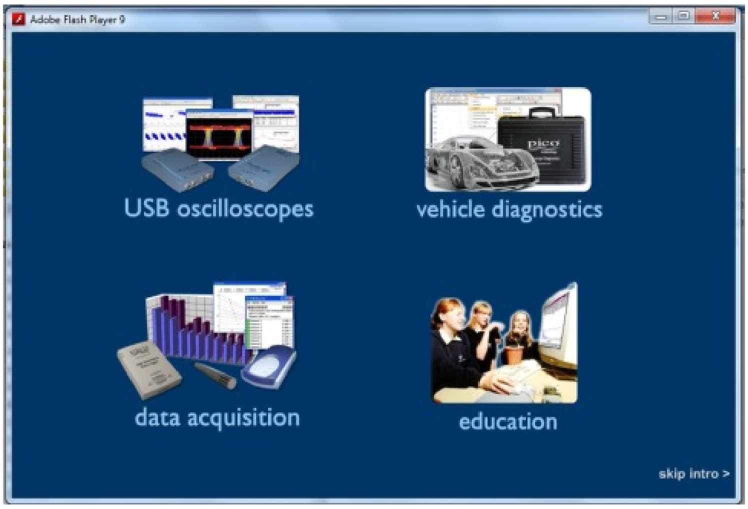
Open folder to view files
using Windows Explorer

View more AutoPlay options in Control Panel

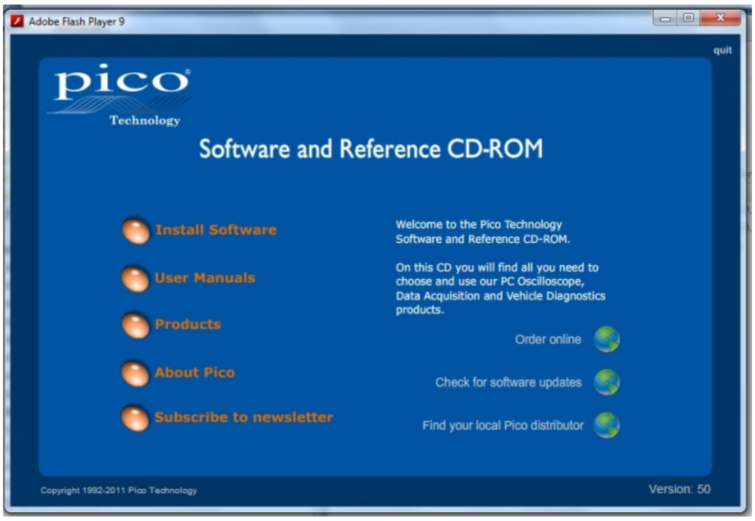
4) Select language menu as “English”, Enter



5) Select “USB oscilloscope”, Enter



6) Select “Install Software”, Enter



Software and Reference CD-ROM

Install Software

User Manuals

Products

About Pico

Subscribe to newsletter

Welcome to the Pico Technology Software and Reference CD-ROM.

On this CD you will find all you need to choose and use our PC Oscilloscope, Data Acquisition and Vehicle Diagnostics products.

Order online

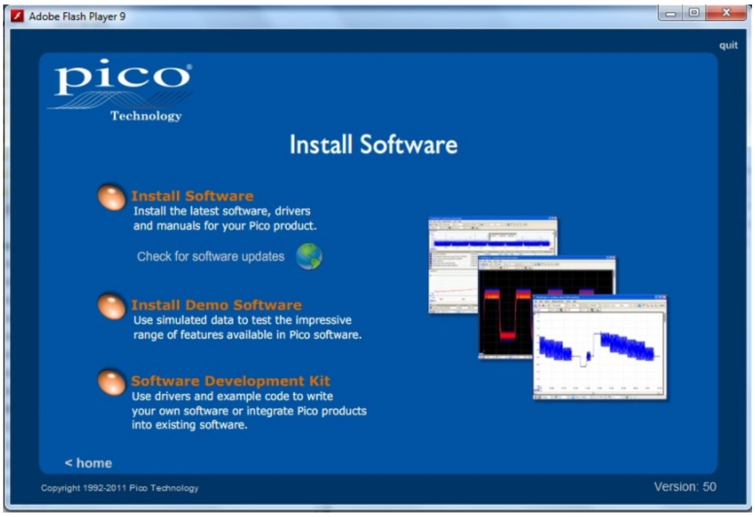
Check for software updates

Find your local Pico distributor

Copyright 1992-2011 Pico Technology

Version: 50

7) At Install Software, select “Install Software”, Enter



Install Software

Install Software
Install the latest software, drivers and manuals for your Pico product.

Check for software updates

Install Demo Software
Use simulated data to test the impressive range of features available in Pico software.

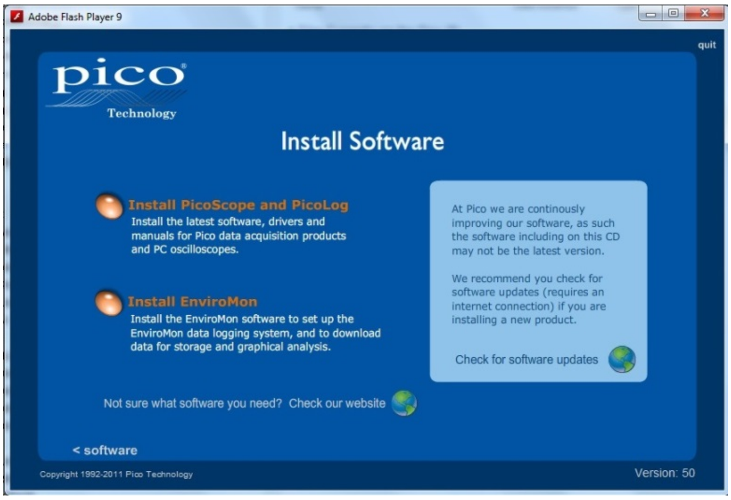
Software Development Kit
Use drivers and example code to write your own software or integrate Pico products into existing software.

< home

Copyright 1992-2011 Pico Technology

Version: 50

8) Select “Install PicoScope and PicoLog”, Enter



Install Software

Install PicoScope and PicoLog
Install the latest software, drivers and manuals for Pico data acquisition products and PC oscilloscopes.

Install EnviroMon
Install the EnviroMon software to set up the EnviroMon data logging system, and to download data for storage and graphical analysis.

Not sure what software you need? Check our website

At Pico we are continously improving our software, as such the software including on this CD may not be the latest version.

We recommend you check for software updates (requires an internet connection) if you are installing a new product.

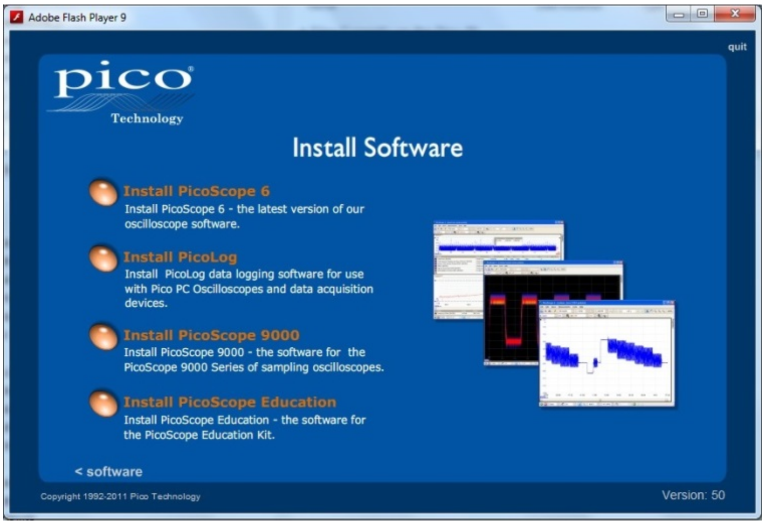
Check for software updates

< software

Copyright 1992-2011 Pico Technology

Version: 50

9) Select “Install PicoScope 6”, Enter



Install Software

Install PicoScope 6
Install PicoScope 6 - the latest version of our oscilloscope software.

Install PicoLog
Install PicoLog data logging software for use with Pico PC Oscilloscopes and data acquisition devices.

Install PicoScope 9000
Install PicoScope 9000 - the software for the PicoScope 9000 Series of sampling oscilloscopes.

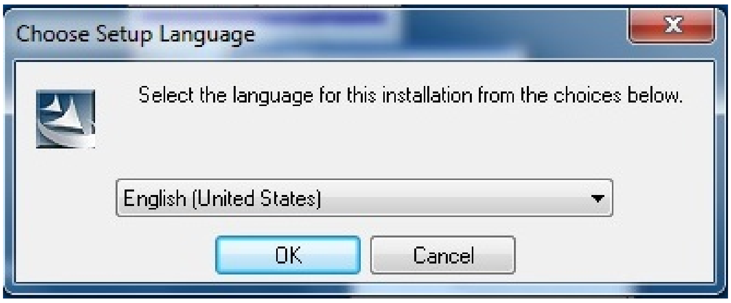
Install PicoScope Education
Install PicoScope Education - the software for the PicoScope Education Kit.

< software

Copyright 1992-2011 Pico Technology

Version: 50

10) Select the language for the installation from the choices given such as “English (United States)”, OK

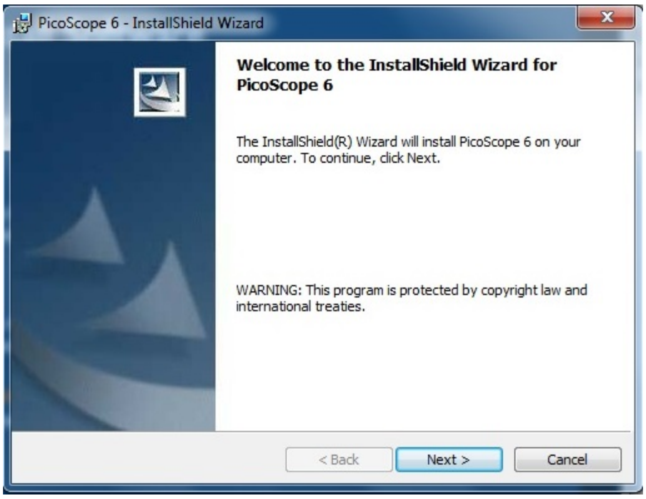


Choose Setup Language

Select the language for this installation from the choices below.

English (United States)

11) At Welcome to the Install Shield Wizard for PicoScope 6, select “Next >”, Enter

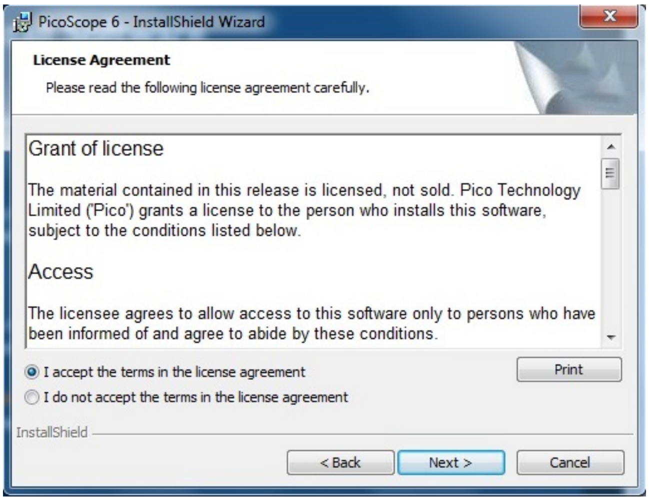


Welcome to the InstallShield Wizard for PicoScope 6

The InstallShield(R) Wizard will install PicoScope 6 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

12) At License Agreement window, select “I accept the terms in the agreement”, Next>, Enter.



License Agreement

Please read the following license agreement carefully.

Grant of license

The material contained in this release is licensed, not sold. Pico Technology Limited ('Pico') grants a license to the person who installs this software, subject to the conditions listed below.

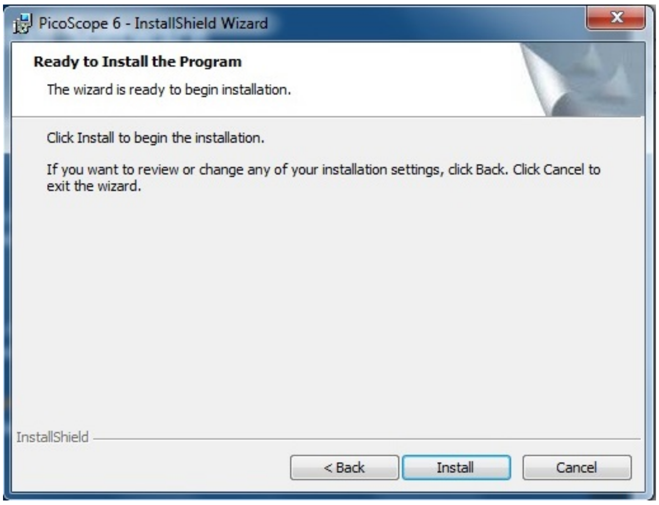
Access

The licensee agrees to allow access to this software only to persons who have been informed of and agree to abide by these conditions.

● I accept the terms in the license agreement

○ I do not accept the terms in the license agreement

13) At Ready to Install the Program, select “Install”, Enter



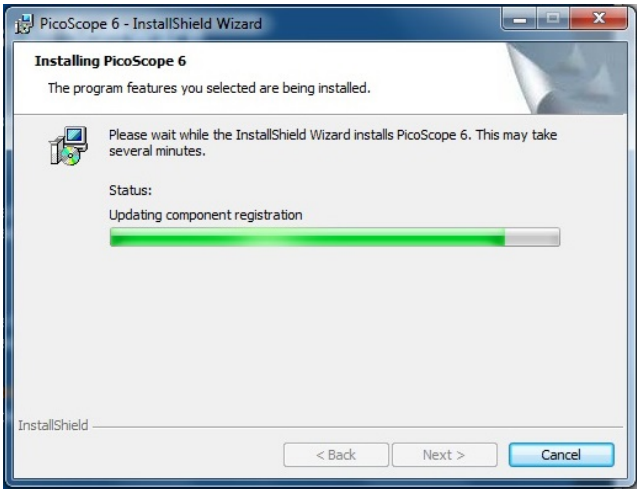
Ready to Install the Program

The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

14) During Installing PicoScope 6, please wait while the installShield Wizard installs PicoScope 6 is finished,


PicoScope 6 - InstallShield Wizard

Installing PicoScope 6

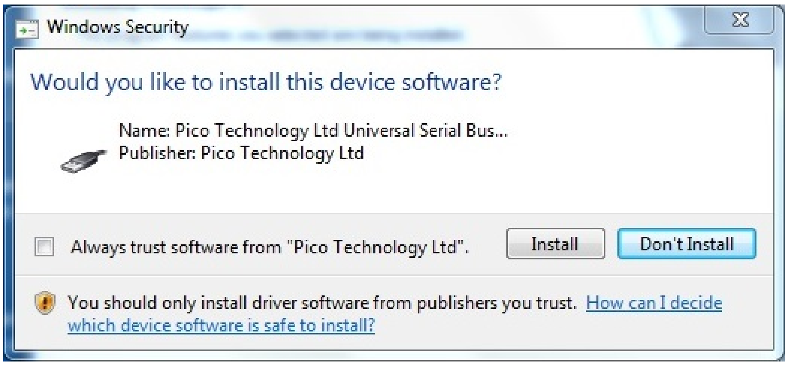
The program features you selected are being installed.

Please wait while the InstallShield Wizard installs PicoScope 6. This may take several minutes.

Status:

Updating component registration

15) On windows Security asking for installing this device software, select "Always trust software from "Pico Technology Ltd.", "Install", Enter


Windows Security

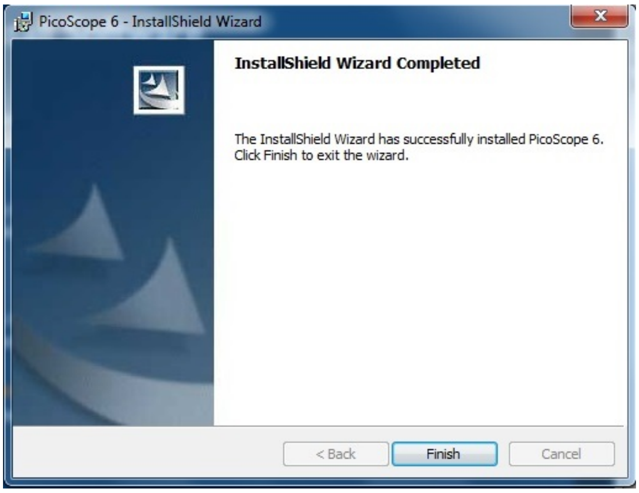
Would you like to install this device software?

Name: Pico Technology Ltd Universal Serial Bus...
Publisher: Pico Technology Ltd

□ Always trust software from "Pico Technology Ltd".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

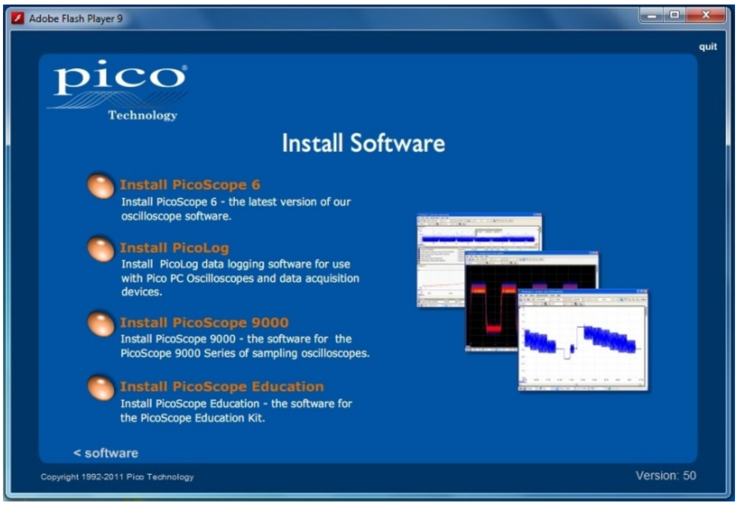
16) At InstallShield Wizard Completed windows, select "Finish", Enter


PicoScope 6 - InstallShield Wizard

InstallShield Wizard Completed

The InstallShield Wizard has successfully installed PicoScope 6. Click Finish to exit the wizard.

17) Select “Quit” on Pico Technology Install Software windows



Install Software

Install PicoScope 6
Install PicoScope 6 - the latest version of our oscilloscope software.

Install PicoLog
Install PicoLog data logging software for use with Pico PC Oscilloscopes and data acquisition devices.

Install PicoScope 9000
Install PicoScope 9000 - the software for the PicoScope 9000 Series of sampling oscilloscopes.

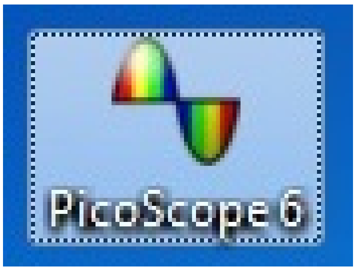
Install PicoScope Education
Install PicoScope Education - the software for the PicoScope Education Kit.

< software

Copyright 1992-2011 Pico Technology

Version: 50

18) After completing the PicoScope 6 installation, the “PicoScope 6” icon will show on the computer display screen



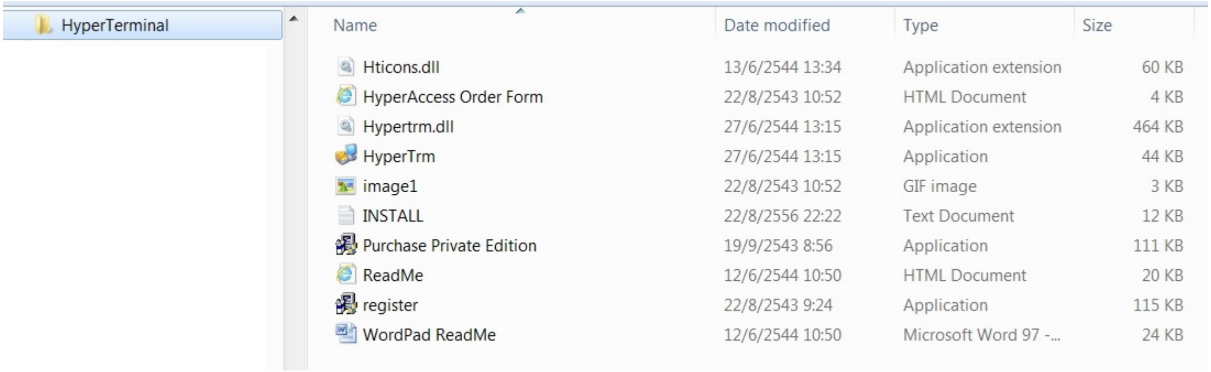
B. Aqua Sound

1) Insert FishFinder Version 3.6 system disk of AquaSound into computer

2) Copy V3b6_FishFinder application to drive c: V3b6_FishFinder application icon will appear on drive c: screen.

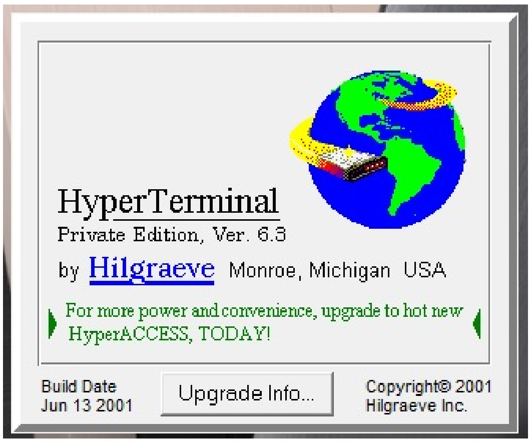
C. <u>Hyper Terminal</u>

1) Load Hyper Terminal software into the computer. Select “HyperTrm” Application, Enter



Name	Date modified	Type	Size
Hticons.dll	13/6/2544 13:34	Application extension	60 KB
HyperAccess Order Form	22/8/2543 10:52	HTML Document	4 KB
Hypertrm.dll	27/6/2544 13:15	Application extension	464 KB
HyperTrm	27/6/2544 13:15	Application	44 KB
image1	22/8/2543 10:52	GIF image	3 KB
INSTALL	22/8/2556 22:22	Text Document	12 KB
Purchase Private Edition	19/9/2543 8:56	Application	111 KB
ReadMe	12/6/2544 10:50	HTML Document	20 KB
register	22/8/2543 9:24	Application	115 KB
WordPad ReadMe	12/6/2544 10:50	Microsoft Word 97 -...	24 KB

2) Hyper Terminal program start-up will appear on display as follows:



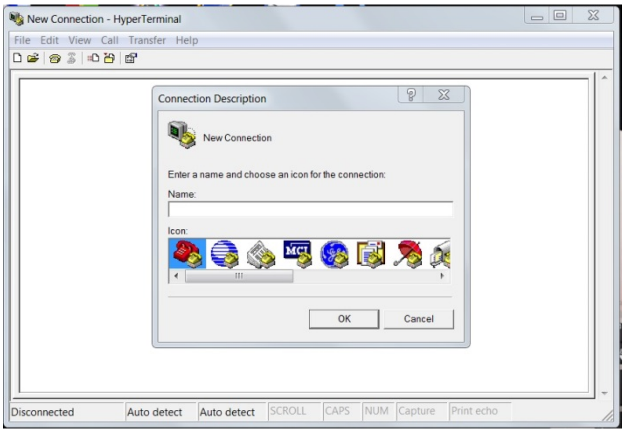
HyperTerminal
Private Edition, Ver. 6.3
by Hilgraeve Monroe, Michigan USA

For more power and convenience, upgrade to hot new HyperACCESS, TODAY!

Build Date
Jun 13 2001

Copyright© 2001
Hilgraeve Inc.

3) Connection Description windows will appear for information selection


New Connection - HyperTerminal

Connection Description

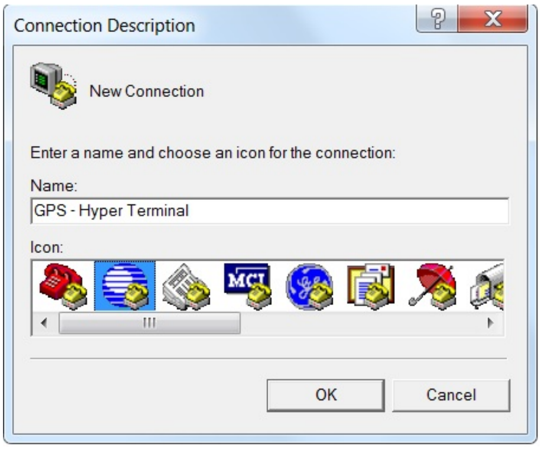
New Connection

Enter a name and choose an icon for the connection:

Name:

Icon:

4) Enter Name of connection, (For example: “GPS – Hyper Terminal”)
Select type of icon for current connection


Connection Description

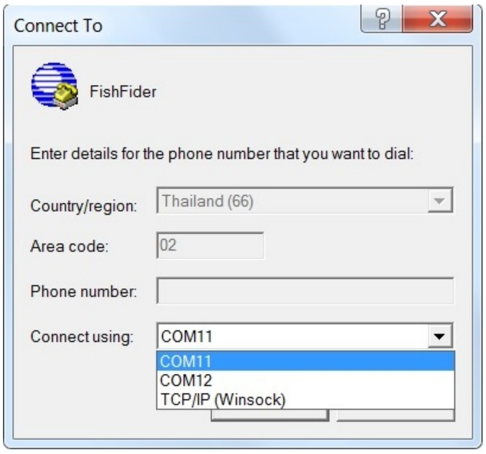
New Connection

Enter a name and choose an icon for the connection:

Name: GPS - Hyper Terminal

Icon:

5) At “Connect To” windows, select “Connect using:” by selecting from available COM port. Available COM port can be checked by following steps:
Control Panel > Hardware and Sound > Device and Printer > Device Manager > Port (COM & LPT).


Connect To

FishFider

Enter details for the phone number that you want to dial:

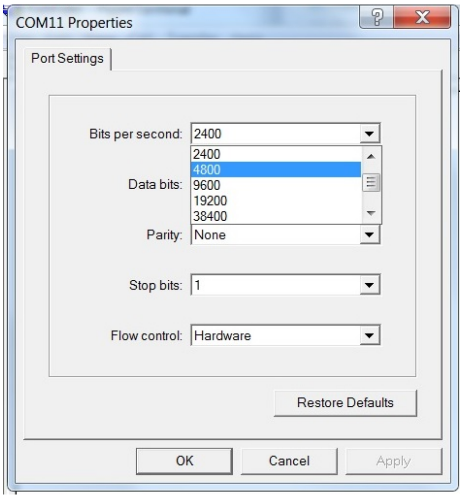
Country/region: Thailand (66)

Area code: 02

Phone number:

Connect using: COM11

6) At “Port Setting” windows, choose “Bits per second:” as 4800.



COM11 Properties

Port Settings

Bits per second: 2400

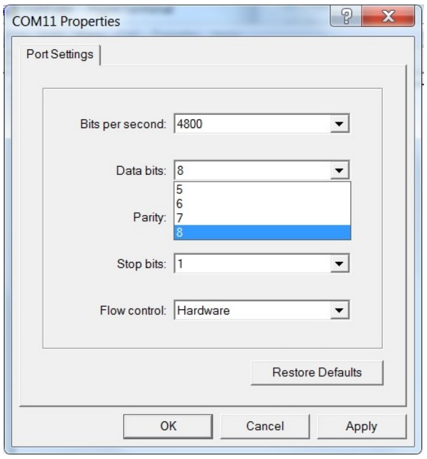
Data bits:

Parity: None

Stop bits: 1

Flow control: Hardware

7) At “Port Setting” windows, choose “Data bits:” as 8



COM11 Properties

Port Settings

Bits per second: 4800

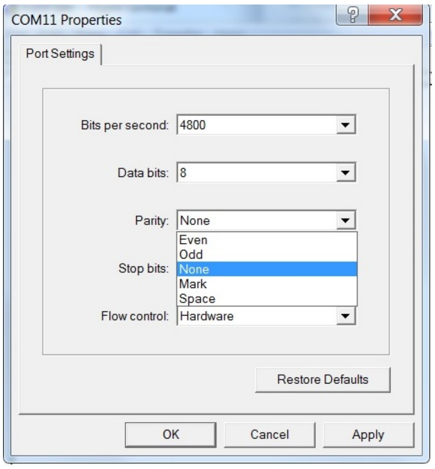
Data bits: 8

Parity:

Stop bits: 1

Flow control: Hardware

8) At “Port Setting” windows, select “Parity:” as None



COM11 Properties

Port Settings

Bits per second: 4800

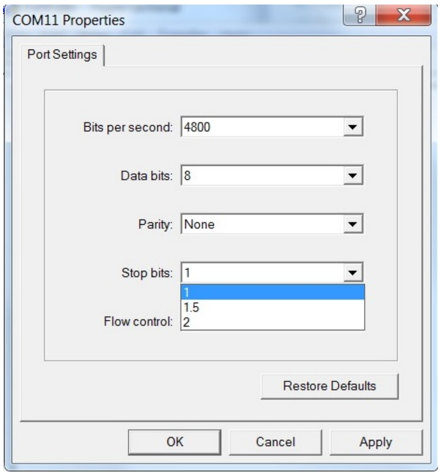
Data bits: 8

Parity: None

Stop bits:

Flow control: Hardware

9) At “Port Setting” windows, select “Stop bits:” as 1



COM11 Properties

Port Settings

Bits per second: 4800

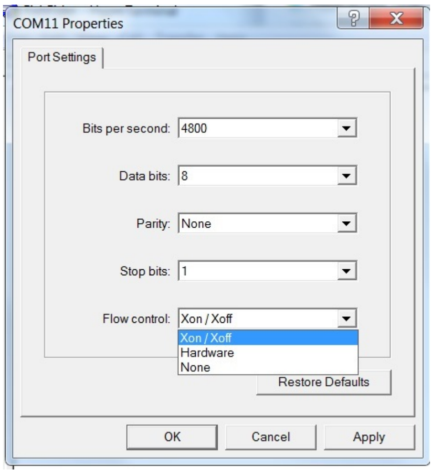
Data bits: 8

Parity: None

Stop bits: 1

Flow control:

10) At “Port Setting” windows, select “Flow control:” as Xon/Xoff



COM11 Properties

Port Settings

Bits per second: 4800

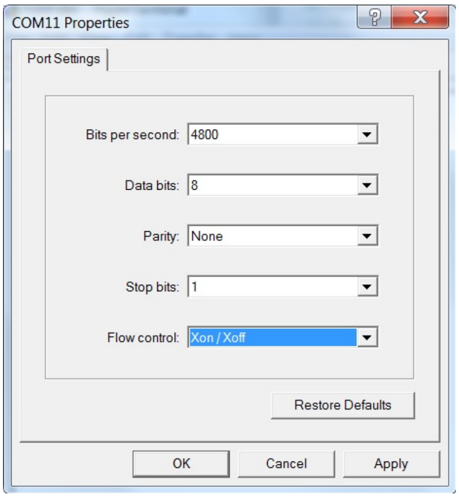
Data bits: 8

Parity: None

Stop bits: 1

Flow control: Xon / Xoff

11) After completing Port Setting,enter OK.



COM11 Properties

Port Settings

Bits per second: 4800

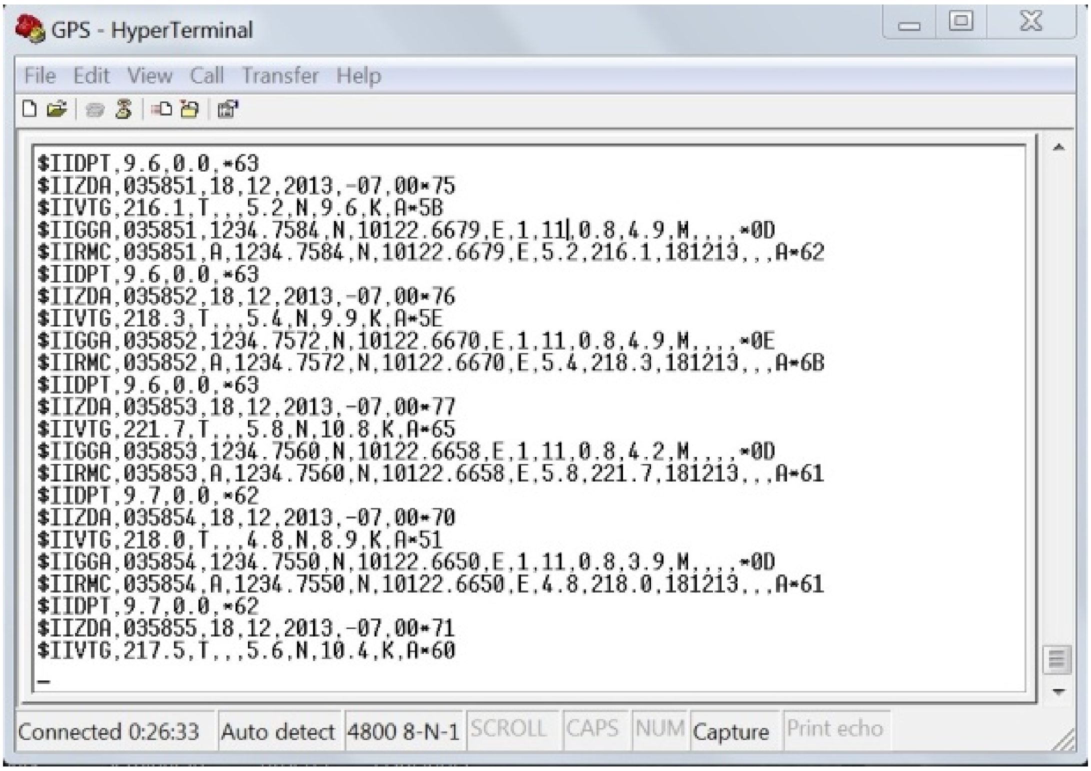
Data bits: 8

Parity: None

Stop bits: 1

Flow control: Xon / Xoff

12) The Hyper Terminal program will start-up with GPS – Hyper Terminal module as already set up.

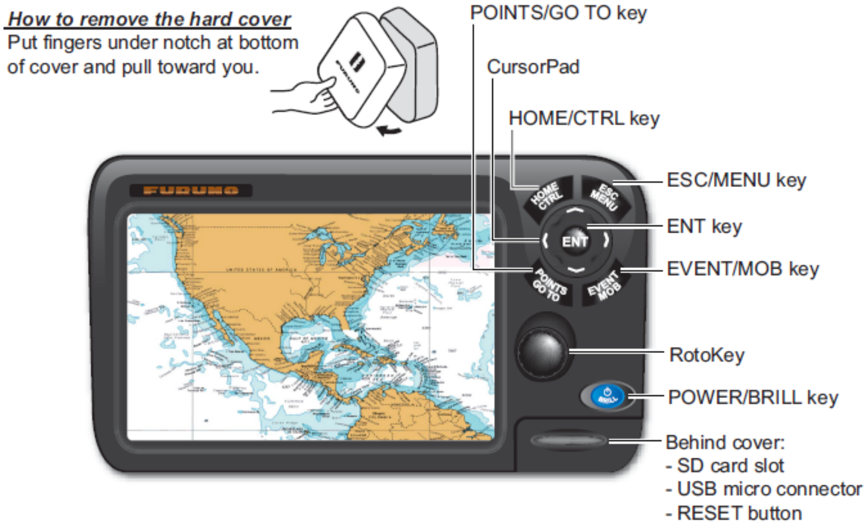


```
$IIDPT,9.6,0.0,*63
$IIZDA,035851,18,12,2013,-07,00*75
$IIVTG,216.1,T,,,5.2,N,9.6,K,A*5B
$IIGGA,035851,1234.7584,N,10122.6679,E,1,11,0.8,4.9,M,,,,*0D
$IIRMC,035851,A,1234.7584,N,10122.6679,E,5.2,216.1,181213,,,A*62
$IIDPT,9.6,0.0,*63
$IIZDA,035852,18,12,2013,-07,00*76
$IIVTG,218.3,T,,,5.4,N,9.9,K,A*5E
$IIGGA,035852,1234.7572,N,10122.6670,E,1,11,0.8,4.9,M,,,,*0E
$IIRMC,035852,A,1234.7572,N,10122.6670,E,5.4,218.3,181213,,,A*6B
$IIDPT,9.6,0.0,*63
$IIZDA,035853,18,12,2013,-07,00*77
$IIVTG,221.7,T,,,5.8,N,10.8,K,A*65
$IIGGA,035853,1234.7560,N,10122.6658,E,1,11,0.8,4.2,M,,,,*0D
$IIRMC,035853,A,1234.7560,N,10122.6658,E,5.8,221.7,181213,,,A*61
$IIDPT,9.7,0.0,*62
$IIZDA,035854,18,12,2013,-07,00*70
$IIVTG,218.0,T,,,4.8,N,8.9,K,A*51
$IIGGA,035854,1234.7550,N,10122.6650,E,1,11,0.8,3.9,M,,,,*0D
$IIRMC,035854,A,1234.7550,N,10122.6650,E,4.8,218.0,181213,,,A*61
$IIDPT,9.7,0.0,*62
$IIZDA,035855,18,12,2013,-07,00*71
$IIVTG,217.5,T,,,5.6,N,10.4,K,A*60
_
```

FURUNO GP-1670F OPERATIONAL PROCEDURE*

1) Turn the Power On, FURUNO GP-1670F

Press ⏻ to turn the power ON.

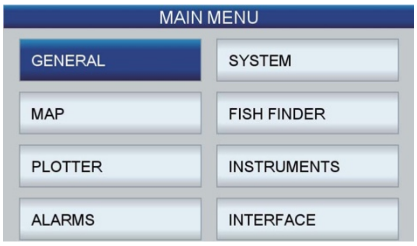


※ This figure is quoted from “FURUNO GPS PLOTTER/SOUNDER GP1670F, GP1870F OPERTOR’S MANUAL” of product GP-1670F of Furuno Electric Co., Ltd.

2) Menu Operation

This section shows how to operate the menu. There are eight menus, [General], [Map], [Plotter], [Alarms], [System], [Fish Finder], [Instruments] and [Interface].

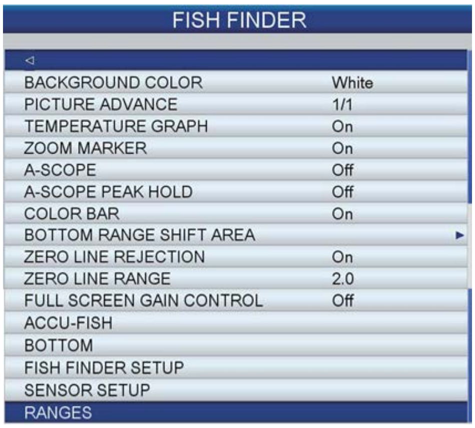
Long-push of the **ESC/MENU** key will show the main menu. Select “FISH FINDER”



MAIN MENU

GENERAL	SYSTEM
MAP	FISH FINDER
PLOTTER	INSTRUMENTS
ALARMS	INTERFACE

3) Select “BOTTOM RANGE SHIFT AREA” for RANGE selection

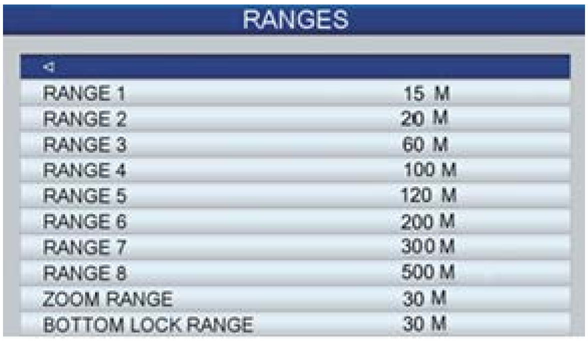


FISH FINDER

◁

BACKGROUND COLOR	White
PICTURE ADVANCE	1/1
TEMPERATURE GRAPH	On
ZOOM MARKER	On
A-SCOPE	Off
A-SCOPE PEAK HOLD	Off
COLOR BAR	On
BOTTOM RANGE SHIFT AREA	►
ZERO LINE REJECTION	On
ZERO LINE RANGE	2.0
FULL SCREEN GAIN CONTROL	Off
ACCU-FISH	
BOTTOM	
FISH FINDER SETUP	
SENSOR SETUP	
RANGES	

[RANGES]: The default range settings are suitable for most applications. However, the ranges can be customized to suit one’s needs, i.e. [RANGE 1] - [RANGE 8]. Set the ranges in descending order. Make sure that each range is lower than its preceding range.



RANGES

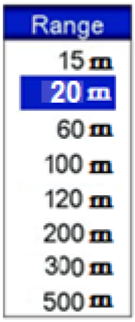
◁

RANGE 1	15 M
RANGE 2	20 M
RANGE 3	60 M
RANGE 4	100 M
RANGE 5	120 M
RANGE 6	200 M
RANGE 7	300 M
RANGE 8	500 M
ZOOM RANGE	30 M
BOTTOM LOCK RANGE	30 M

Select a display range

Range can be selected automatically or manually. Open the RotoKey menu then select [Auto Range] and [Auto] or [Manual].

Select manual operation, the [RANGE] window, which appears on the right. Turn the key clockwise to increase the range and counterclockwise to decrease the range. Select the Range at 20 m.



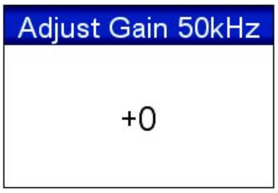
Range

15 m
20 m
60 m
100 m
120 m
200 m
300 m
500 m

4) Adjust the Gain

The gain controls how echoes of different strengths are displayed. Gain is automatically adjusted; however, the gain can be fine tuned accordingly to meet local characteristics, etc. Set the gain to show a slight amount of noise on the screen. Increase the gain for greater depths and lower the gain for shallow waters.

To adjust the gain, open the RotoKey menu then select [Gain 50k] or [Gain 200k] followed by the soft control labeled with the frequency that is desired to be adjusted, then the corresponding adjustment window appears. Rotate the key clockwise to increase the gain, counterclockwise to decrease it.

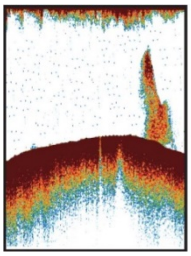


Adjust Gain 50kHz

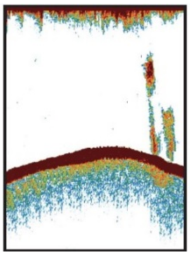
+0

How to adjust the gain

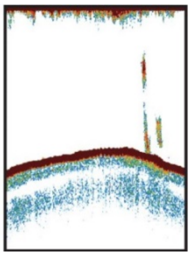
Since the gain controls how echoes of different strengths are displayed, set the gain to show a slight amount of noise on the screen. Increase the gain for greater depths and lower the gain for shallow waters.



Gain too high



Gain proper



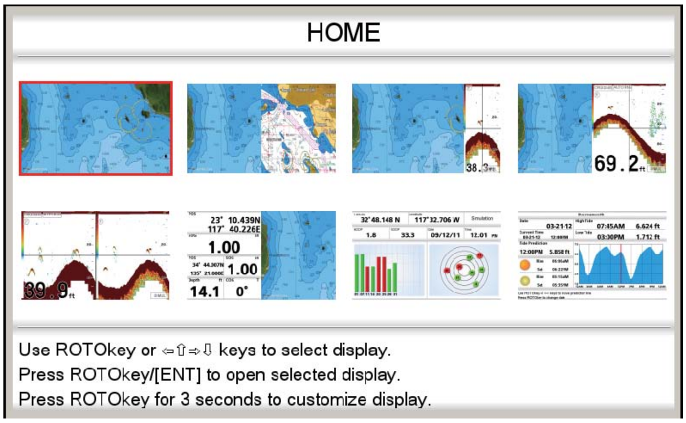
Gain too low

CAUTION

Adjust the gain correctly. Incorrect adjustment can lead to a dangerous situation especially if the boat is steered according to the depth indication.

5) Home Screen (Display Selection)

Select a display from the home screen which has eight displays. Press the HOME/CTRL key to show the home screen. Operate the CursorPad or rotate the RotoKey™ to select a display. The current selection is circumscribed by a red rectangle. Press the RotoKey™ or ENT key to confirm your selection.

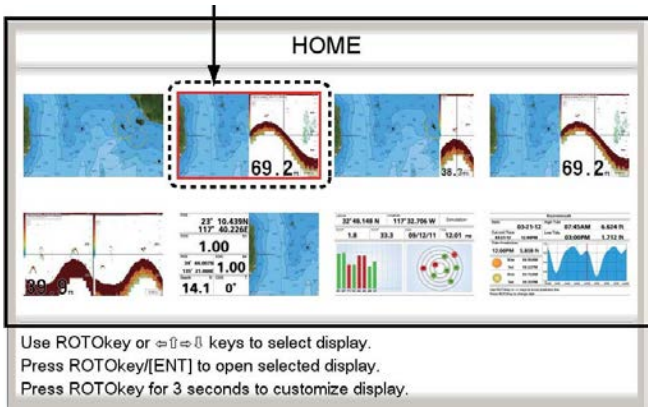


HOME

Use ROTOkey or ⇦⇧⇨⇩ keys to select display.
Press ROTOkey/[ENT] to open selected display.
Press ROTOkey for 3 seconds to customize display.

Select a display for the right half then push RotoKey™. For example, select the fish finder display. Control returns to the home screen. The result of your selection appears on the home screen.

Plotter, fish finder display



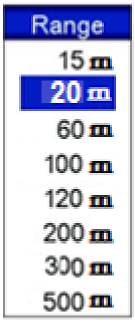
HOME

Use ROTOkey or ⇦⇧⇨⇩ keys to select display.
Press ROTOkey/[ENT] to open selected display.
Press ROTOkey for 3 seconds to customize display.

DATA COLLECTION OPERATION PROCEDURE

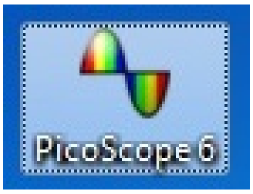
1) Turn on FURUNO GP-1670F

   - Select the Range at 20 m.



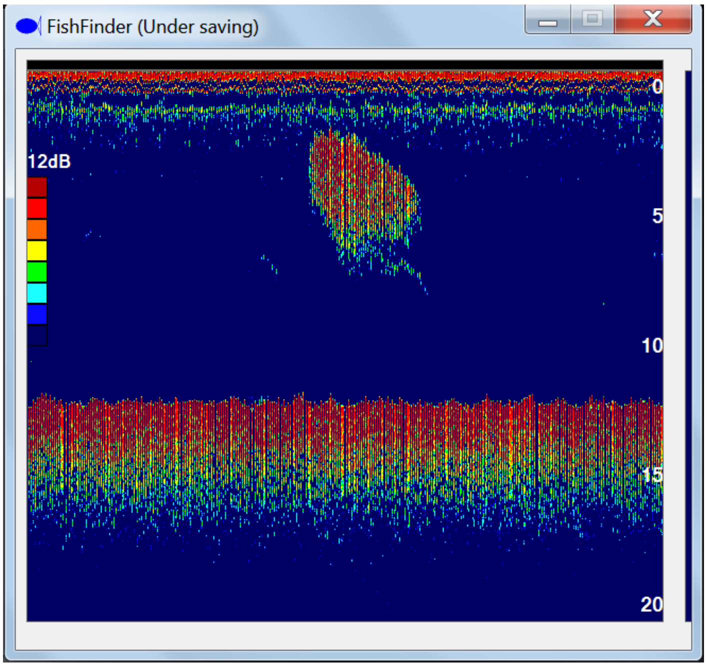
2) Turn on Computer with Windows 7

3) Start PicoScope 6

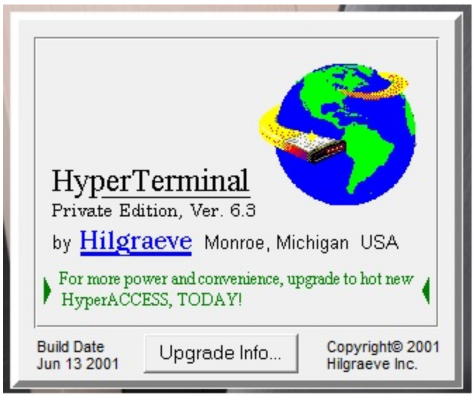


4) Start FishFinder V.3.6

When FishFinder program is running, a window on monitor screen will display an echogram which is indicated by water depth range and echo color scale.



5) Start Hyper Terminal: (GPS – Hyper Terminal)



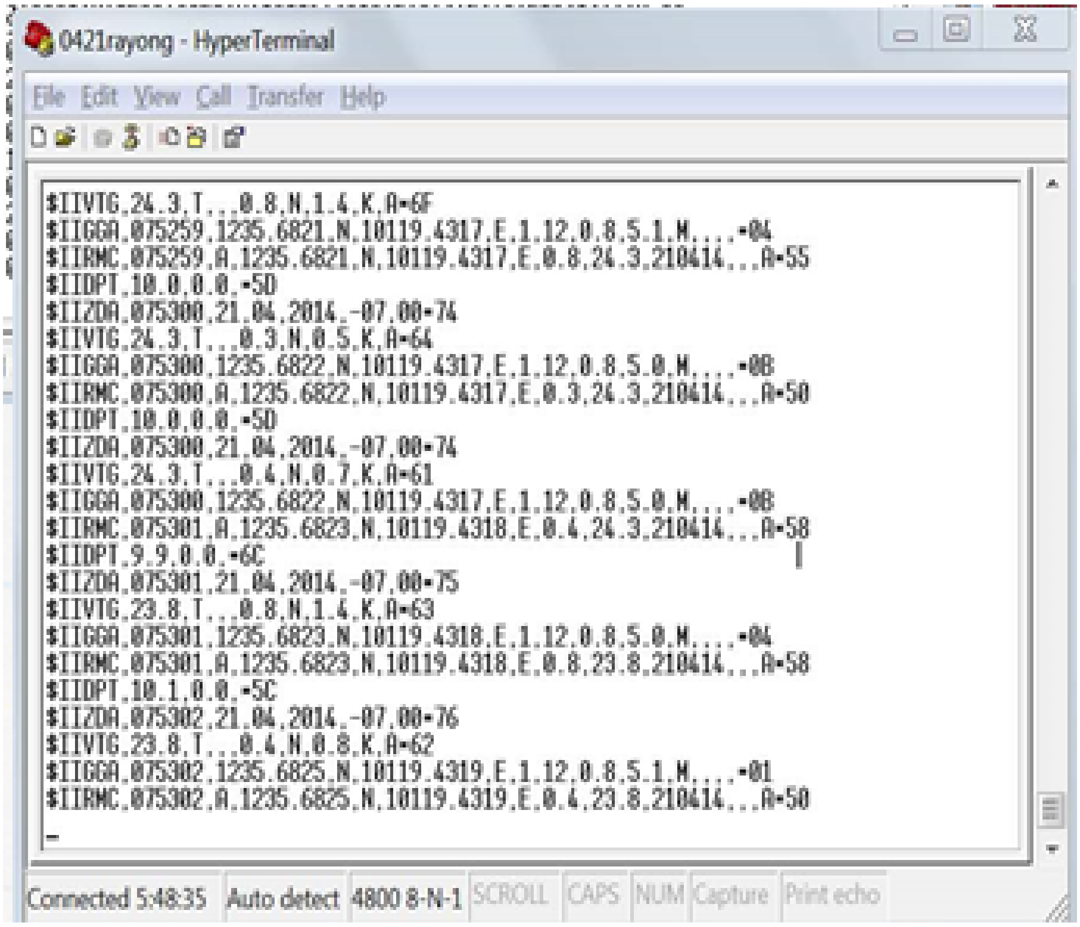
HyperTerminal
Private Edition, Ver. 6.3
by Hilgraeve Monroe, Michigan USA

For more power and convenience, upgrade to hot new HyperACCESS, TODAY!

Build Date
Jun 13 2001

Copyright© 2001
Hilgraeve Inc.

When Hyper Terminal program is running, a window on monitor screen will display echo sounder information in various format of terminal information.

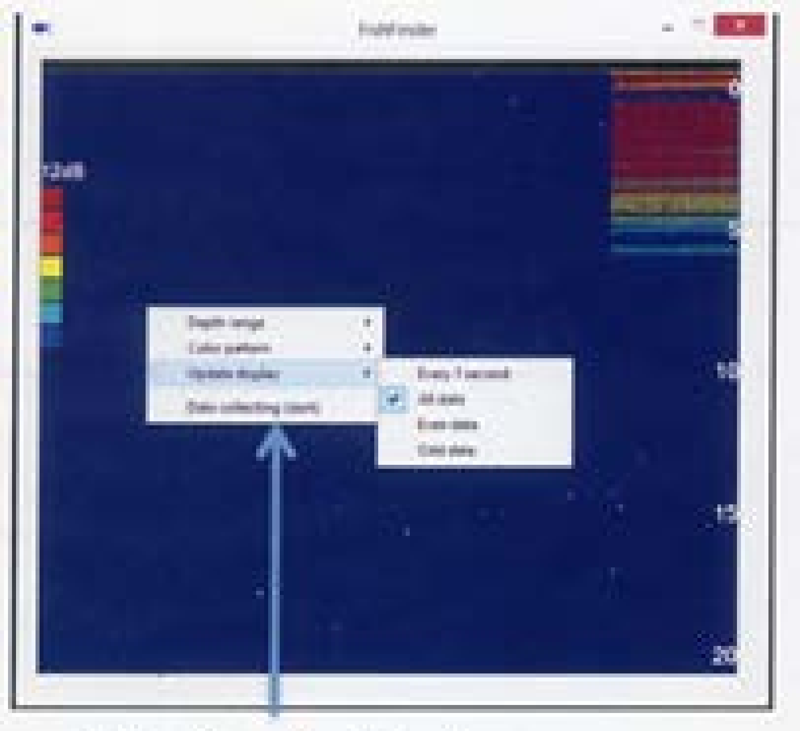


```
$IIVTG,24.3,T,,,0.8,N,1.4,K,A*6F
$IIGGA,075259,1235.6821,N,10119.4317,E,1,12,0.8,5.1,M,,,,*04
$IIRMC,075259,A,1235.6821,N,10119.4317,E,0.8,24.3,210414,,,A*55
$IIDPT,10.0,0.0,*5D
$IIZDA,075300,21,04,2014,-07,00*74
$IIVTG,24.3,T,,,0.3,N,0.5,K,A*64
$IIGGA,075300,1235.6822,N,10119.4317,E,1,12,0.8,5.0,M,,,,*0B
$IIRMC,075300,A,1235.6822,N,10119.4317,E,0.3,24.3,210414,,,A*50
$IIDPT,10.0,0.0,*5D
$IIZDA,075300,21,04,2014,-07,00*74
$IIVTG,24.3,T,,,0.4,N,0.7,K,A*61
$IIGGA,075300,1235.6822,N,10119.4317,E,1,12,0.8,5.0,M,,,,*0B
$IIRMC,075301,A,1235.6823,N,10119.4318,E,0.4,24.3,210414,,,A*58
$IIDPT,9.9,0.0,*6C
$IIZDA,075301,21,04,2014,-07,00*75
$IIVTG,23.8,T,,,0.8,N,1.4,K,A*63
$IIGGA,075301,1235.6823,N,10119.4318,E,1,12,0.8,5.0,M,,,,*04
$IIRMC,075301,A,1235.6823,N,10119.4318,E,0.8,23.8,210414,,,A*58
$IIDPT,10.1,0.0,*5C
$IIZDA,075302,21,04,2014,-07,00*76
$IIVTG,23.8,T,,,0.4,N,0.8,K,A*62
$IIGGA,075302,1235.6825,N,10119.4319,E,1,12,0.8,5.1,M,,,,*01
$IIRMC,075302,A,1235.6825,N,10119.4319,E,0.4,23.8,210414,,,A*50
_
```

Connected 5:48:35 | Auto detect | 4800 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo

6) Echo sounder data collection

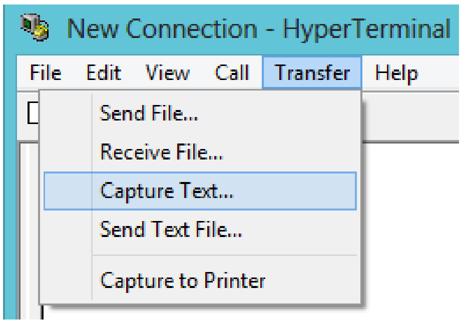
- Move mouse cursor into the echogram window and use right bottom mouse click for data collection drop down menu.
- Select “Collecting Start” button, left bottom mouse click. The echo sounder data will automatically create new data recorded file using current date and time as file name on Desktop folder. The format of file name will start with “FishFinder” followed by Date such as “FishFinder_yyyymmdd_hhmmss.csv”.
- To stop data recording, select “Collecting Stop” button, and use left bottom mouse click.

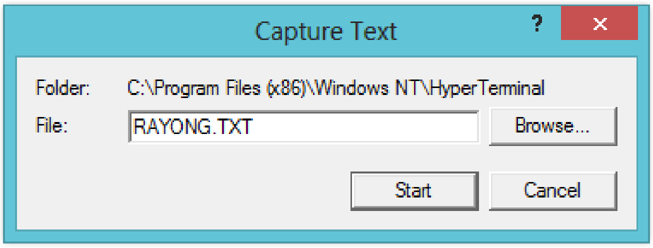


Collecting start button

7) Ship position data collection

- In Hyper Terminal window, select drop down menu of “Transfer”, Capture Text” and enter Capture Text folder and file name.
- Select “Start” button. The echo sounder information text will automatically be recorded into defined data file name.

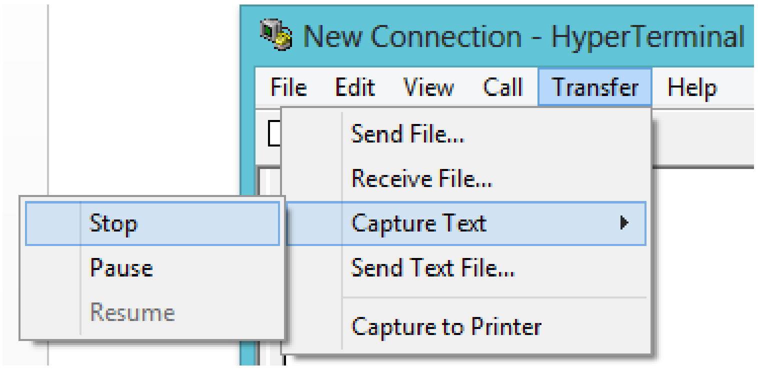




Folder: C:\Program Files (x86)\Windows NT\HyperTerminal

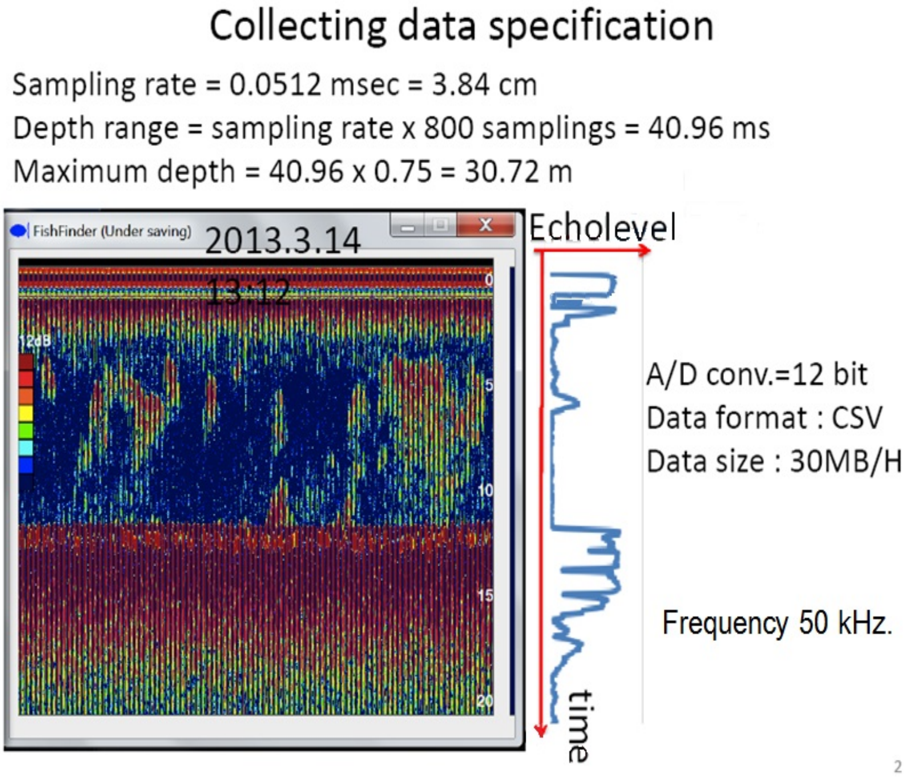
File: RAYONG.TXT

- To stop data recording, select “Transfer”, Capture Text” and “Stop”



ECHO SOUNDER DATA ANALYSIS PROCEDURE

The Echo Sounder Data Collection System (AQFI-1301) was designed to display echogram (Figure 17) and data recorded digital echo signal (50 kHz) into PC screen and hard disc. The Echo Sounder Data Collection System was set up at fixed sampling rate of 0.0512 msec with total 800 sampling echo data of digital value echo-signal recorded into the PC hard disc for each transmitting ping.


Collecting data specification

Sampling rate = 0.0512 msec = 3.84 cm

Depth range = sampling rate x 800 samplings = 40.96 ms

Maximum depth = 40.96 x 0.75 = 30.72 m

A/D conv.=12 bit

Data format : CSV

Data size : 30MB/H

Frequency 50 kHz.

Figure 17. Echogram and A-scope display pattern shown on PC screen.

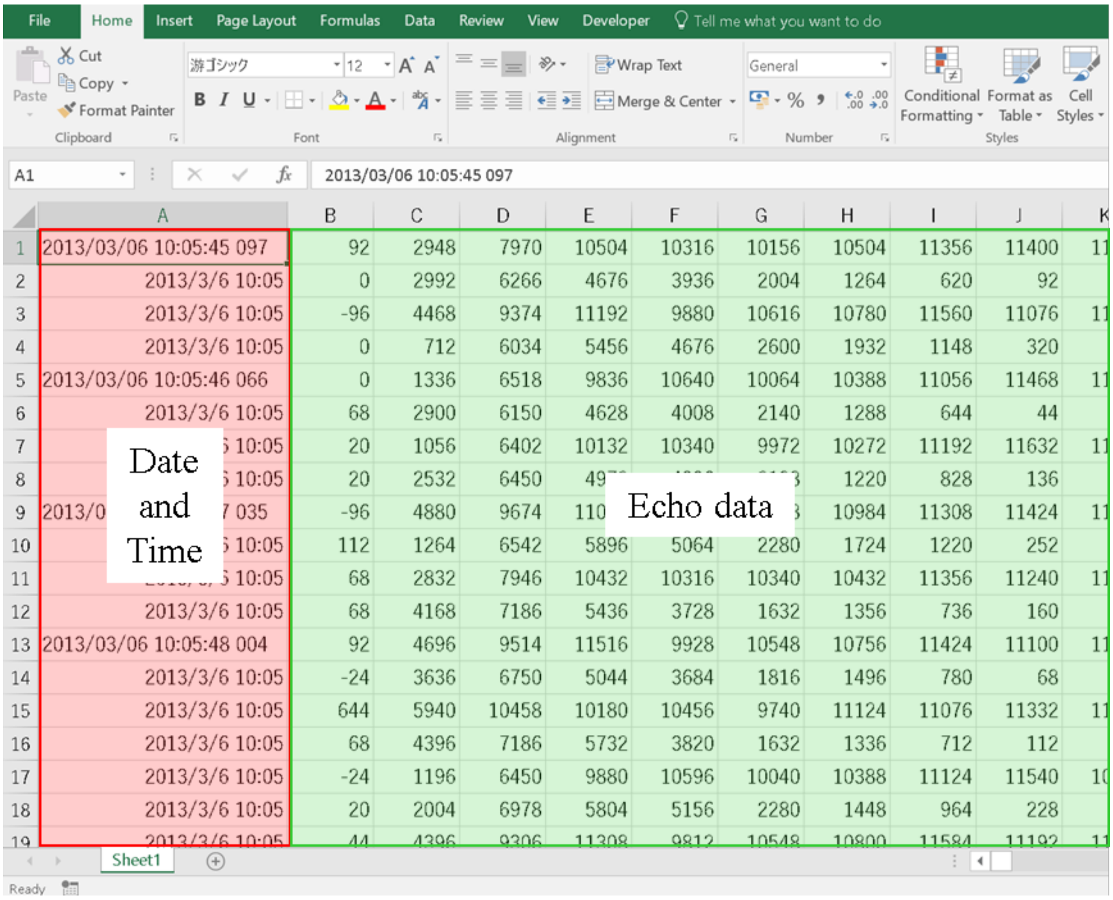
The echo sounder data can be analyzed using Excel program. This manual shows the procedure of analyzing data with Microsoft Excel.

1) Open echo sounder raw data file using the Excel program, then a table like Figure 18 is displated. Column A indicates the recording date and time. The echo sounder data are set from column B to column ADU (800 data). Since the sampling rate of the system is 0.512 msec (and sampling frequency is 19.53 kHz), the echo data are recorded with 3.84 cm resolution as shown by the following equation:

   Resolution = Sampling Rate × Sound Speed / 2
   = 0.512 (msec) × 1500 (m/sec) / 2 = 3.84 (cm)

   And the maximum depth that the system can measure is:

   Max. depth = Resolution × Amount of Data
   = 3.84 cm × 800= 30.72 m



A1 | 2013/03/06 10:05:45 097

	A	B	C	D	E	F	G	H	I	J	K
1	2013/03/06 10:05:45 097	92	2948	7970	10504	10316	10156	10504	11356	11400	11
2	2013/3/6 10:05	0	2992	6266	4676	3936	2004	1264	620	92	
3	2013/3/6 10:05	-96	4468	9374	11192	9880	10616	10780	11560	11076	11
4	2013/3/6 10:05	0	712	6034	5456	4676	2600	1932	1148	320	
5	2013/03/06 10:05:46 066	0	1336	6518	9836	10640	10064	10388	11056	11468	11
6	2013/3/6 10:05	68	2900	6150	4628	4008	2140	1288	644	44	
7	10:05	20	1056	6402	10132	10340	9972	10272	11192	11632	11
8	10:05	20	2532	6450	49		3	1220	828	136	
9	2013/0 7 035	-96	4880	9674	110		3	10984	11308	11424	11
10	10:05	112	1264	6542	5896	5064	2280	1724	1220	252	
11	10:05	68	2832	7946	10432	10316	10340	10432	11356	11240	11
12	2013/3/6 10:05	68	4168	7186	5436	3728	1632	1356	736	160	
13	2013/03/06 10:05:48 004	92	4696	9514	11516	9928	10548	10756	11424	11100	11
14	2013/3/6 10:05	-24	3636	6750	5044	3684	1816	1496	780	68	
15	2013/3/6 10:05	644	5940	10458	10180	10456	9740	11124	11076	11332	11
16	2013/3/6 10:05	68	4396	7186	5732	3820	1632	1336	712	112	
17	2013/3/6 10:05	-24	1196	6450	9880	10596	10040	10388	11124	11540	10
18	2013/3/6 10:05	20	2004	6978	5804	5156	2280	1448	964	228	
19	2013/3/6 10:05	44	4396	9306	11308	9812	10548	10800	11584	11192	11

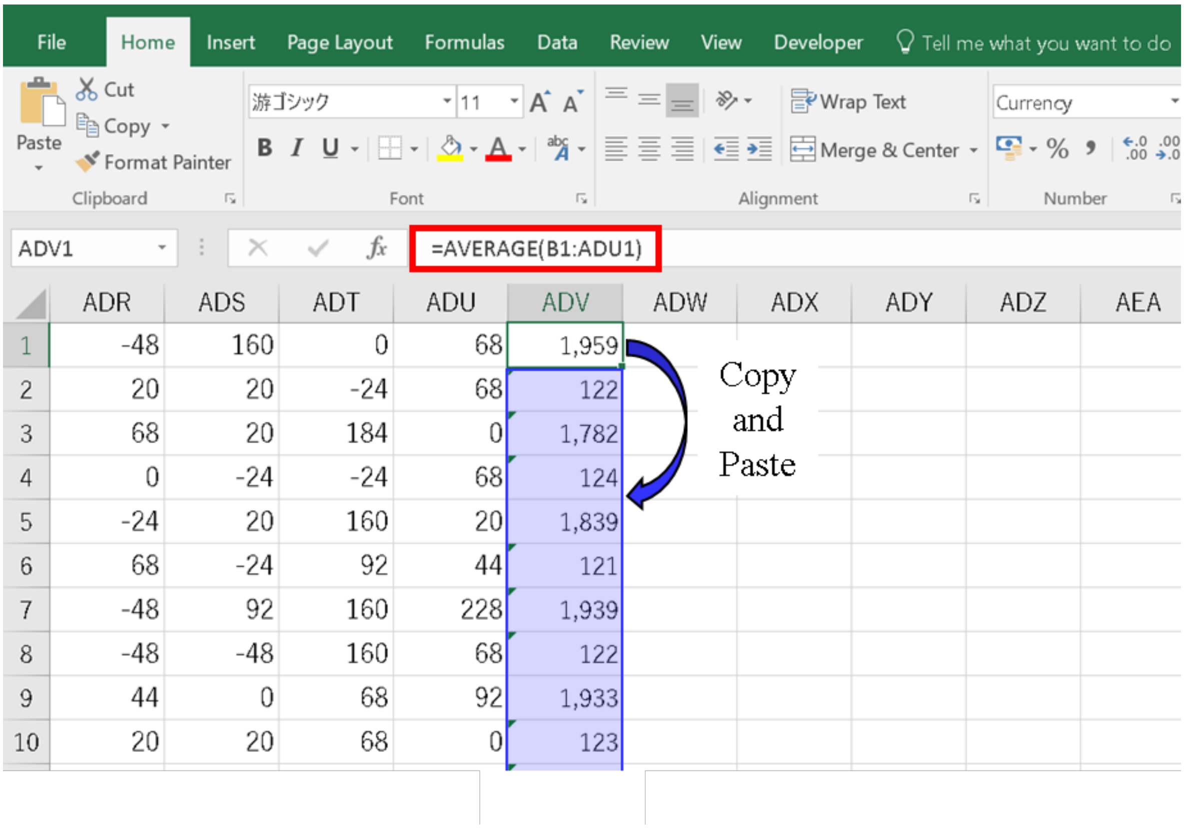
Figure 18. Echo sounder raw data.

The echo sounder FURUNO GP-1670F sends pulses of 50 and 200 kHz alternately. Then, recording echoes are also arranged alternately by 50 and 200 kHz data in the direction of the row. The return echo signal recorded by Echo Sounder Data Collection System (AQFI-1301) on the ".csv" Excel file will also be composed of rows of digital data information of 50 and 200 kHz alternately. However, Echo Sounder Data Collection

System (AQFI-1301) is designed to analyze only 50-kHz data. The following processes show the way to filter the 50-kHz data.

2) Input "=AVERAGE(B1:ADU1)" in cell ADV1 for calculating the average value from cell B1 to ADU1(Figure 19).

3) Copy cell ADV1. Select the whole column ADV and paste the formula (Figure 19). Then the low and high average values are being alternately shown in column ADV.



ADV1 | =AVERAGE(B1:ADU1)

	ADR	ADS	ADT	ADU	ADV	ADW	ADX	ADY	ADZ	AEA
1	-48	160	0	68	1,959					
2	20	20	-24	68	122					
3	68	20	184	0	1,782					
4	0	-24	-24	68	124					
5	-24	20	160	20	1,839					
6	68	-24	92	44	121					
7	-48	92	160	228	1,939					
8	-48	-48	160	68	122					
9	44	0	68	92	1,933					
10	20	20	68	0	123					

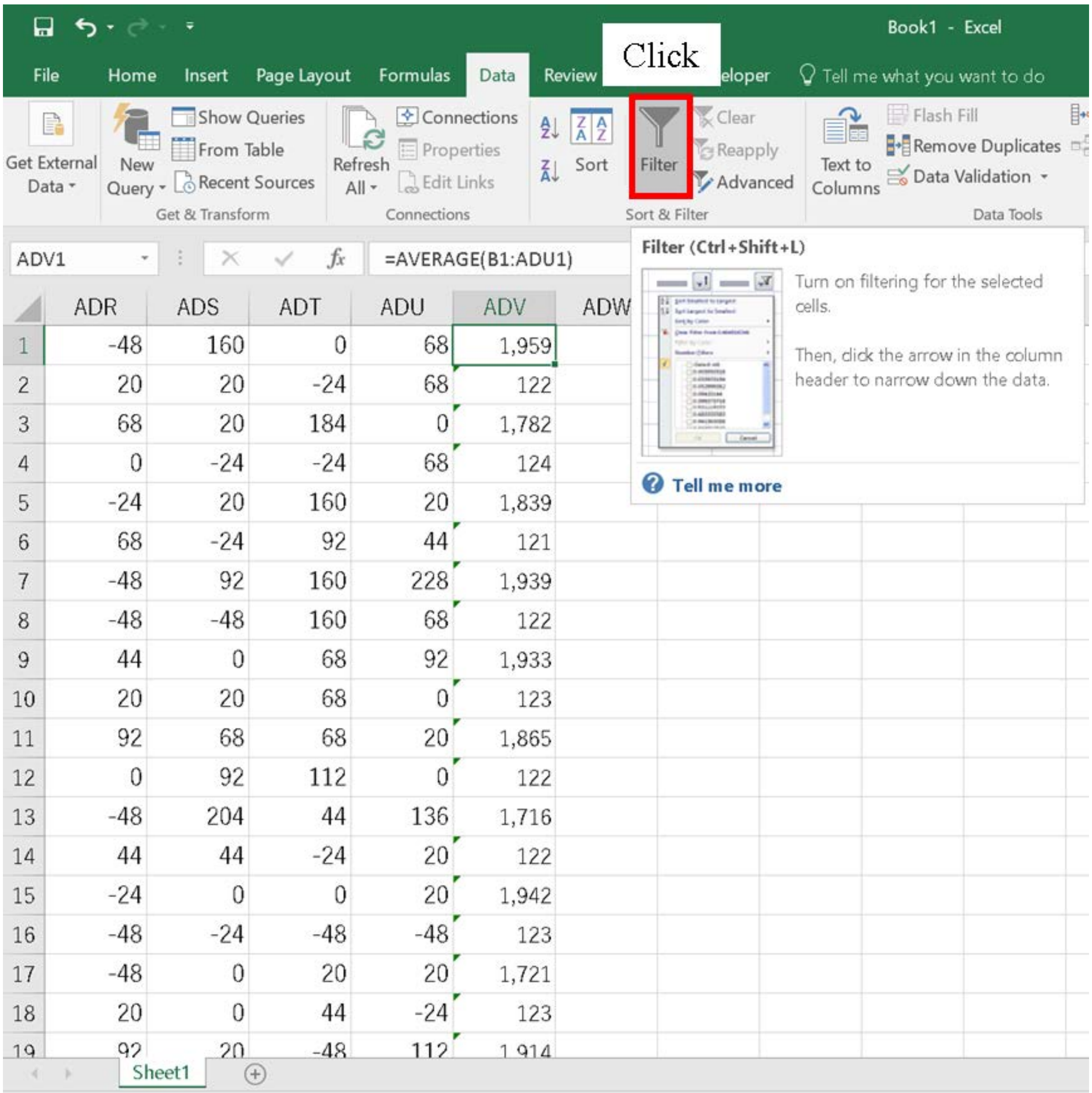
Copy and Paste

Figure 19. Process to average echo data.

The values of 50-kHz data are relatively much higher than those of 200-kHz data. 50 kHz data can be extracted by the "Filter" function.

4) Select cell ADV1 and click "Data" > "Filter" of the upper menu (Figure 20).

5) Click the arrow in column ADV1, and then click "Number Filters". Select "Greater Than Or Equal To…" (Figure 21), then the Custom Auto Filter box is opened. In the Custom AutoFilter box, type the criteria for filtering your data. For example, type 500 in the upper-right box to show the data higher than 500. Click "OK" to apply the filter (Figure 22).



ADV1 =AVERAGE(B1:ADU1)

	ADR	ADS	ADT	ADU	ADV	ADW
1	-48	160	0	68	1,959	
2	20	20	-24	68	122	
3	68	20	184	0	1,782	
4	0	-24	-24	68	124	
5	-24	20	160	20	1,839	
6	68	-24	92	44	121	
7	-48	92	160	228	1,939	
8	-48	-48	160	68	122	
9	44	0	68	92	1,933	
10	20	20	68	0	123	
11	92	68	68	20	1,865	
12	0	92	112	0	122	
13	-48	204	44	136	1,716	
14	44	44	-24	20	122	
15	-24	0	0	20	1,942	
16	-48	-24	-48	-48	123	
17	-48	0	20	20	1,721	
18	20	0	44	-24	123	
19	92	20	-48	112	1,914	

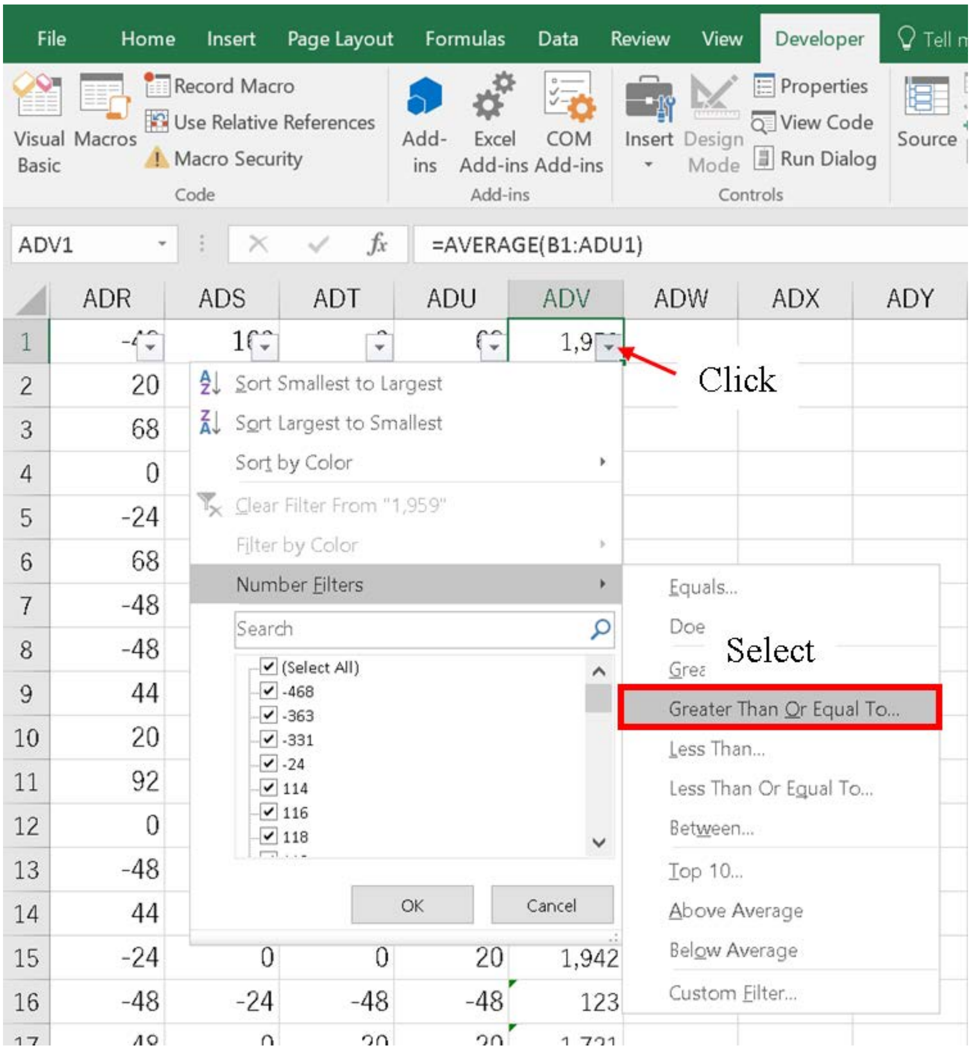
Filter (Ctrl+Shift+L)

Turn on filtering for the selected cells.

Then, click the arrow in the column header to narrow down the data.

Tell me more

Figure 20. Activation of the “Filter” tool.

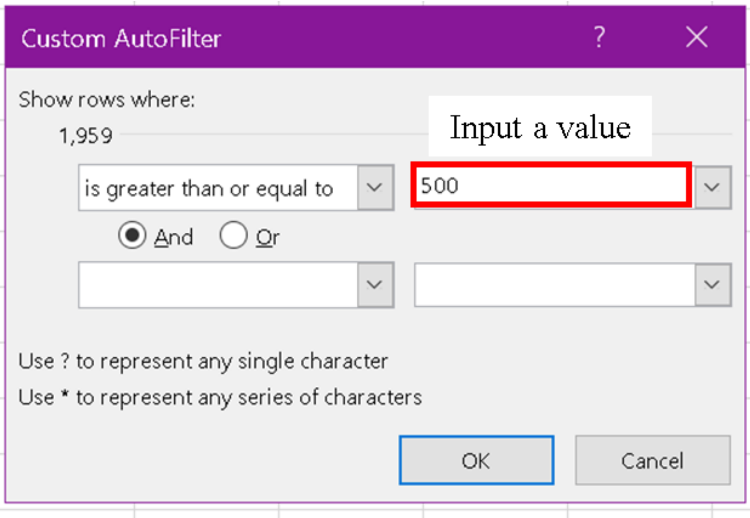


ADV1 =AVERAGE(B1:ADU1)

Click

Select

Figure 21. Selecting the way of filtering data.



Show rows where:

1,959

is greater than or equal to 500

● And ○ Or

Use ? to represent any single character

Use * to represent any series of characters

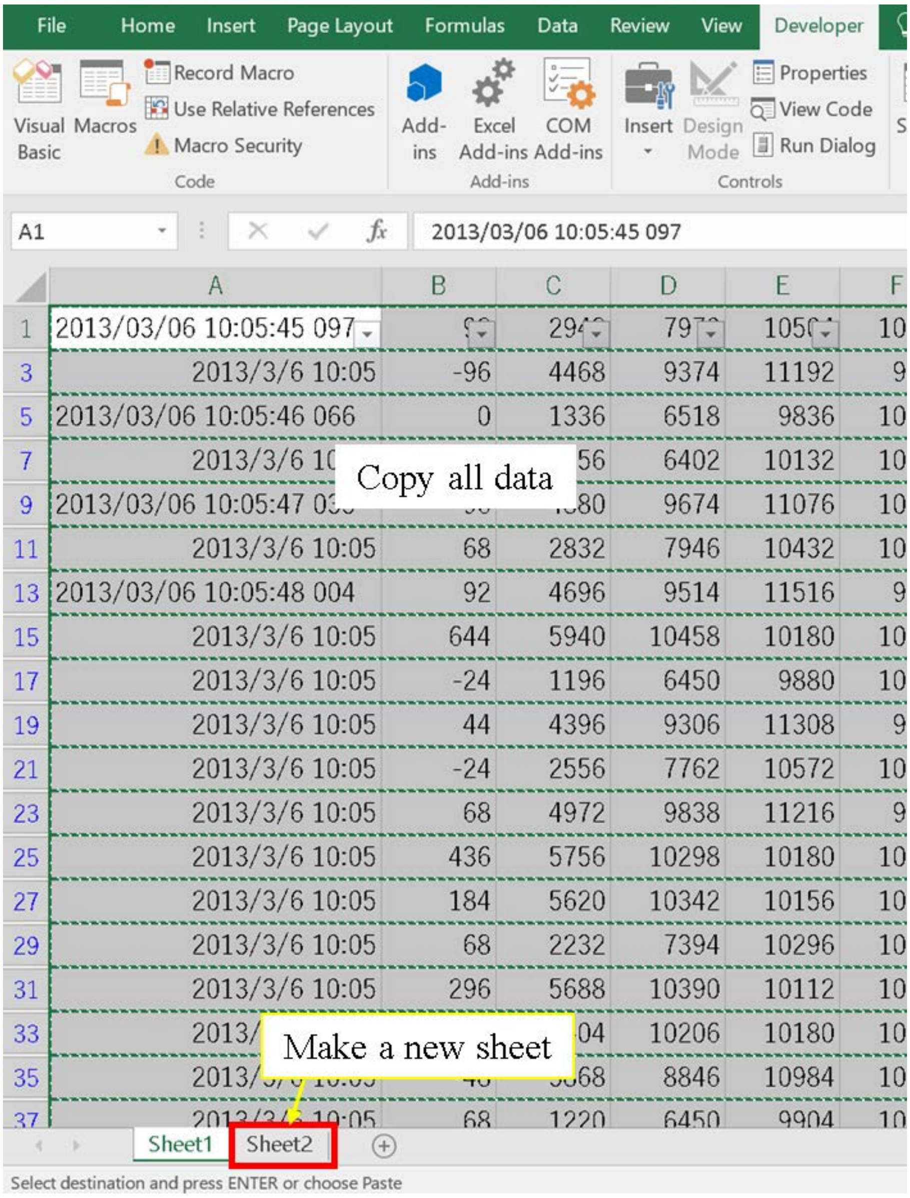
Input a value

Figure 22. Custom AutoFilter box.

6) Copy the filtered data (column A to ADV) and paste to a new sheet (the sheet is written as “Sheet 2” bellow) (Figure 23).

   The data should be converted to voltage values to evaluate the echo data. The equation for converting is:
   VR = 0.00013 × Raw data value

7) Make an additional new sheet (the sheet is written as “Sheet 3” below). Copy column A of “Sheet 2” and paste to column A of “Sheet 3”. Select cell B1 of “Sheet 3” and input “=Sheet2!B1*0.00013” (Figure 24). Sheet 2 should be renamed, then input the name of the “renamed Sheet 2” instead of just Sheet 2 in the equation. Copy cell B1 of “Sheet 3”. Select cell B1 to cell ADU *x* (where *x* is the number of rows entered in the 50-kHz data in “Sheet 2”), and then paste.

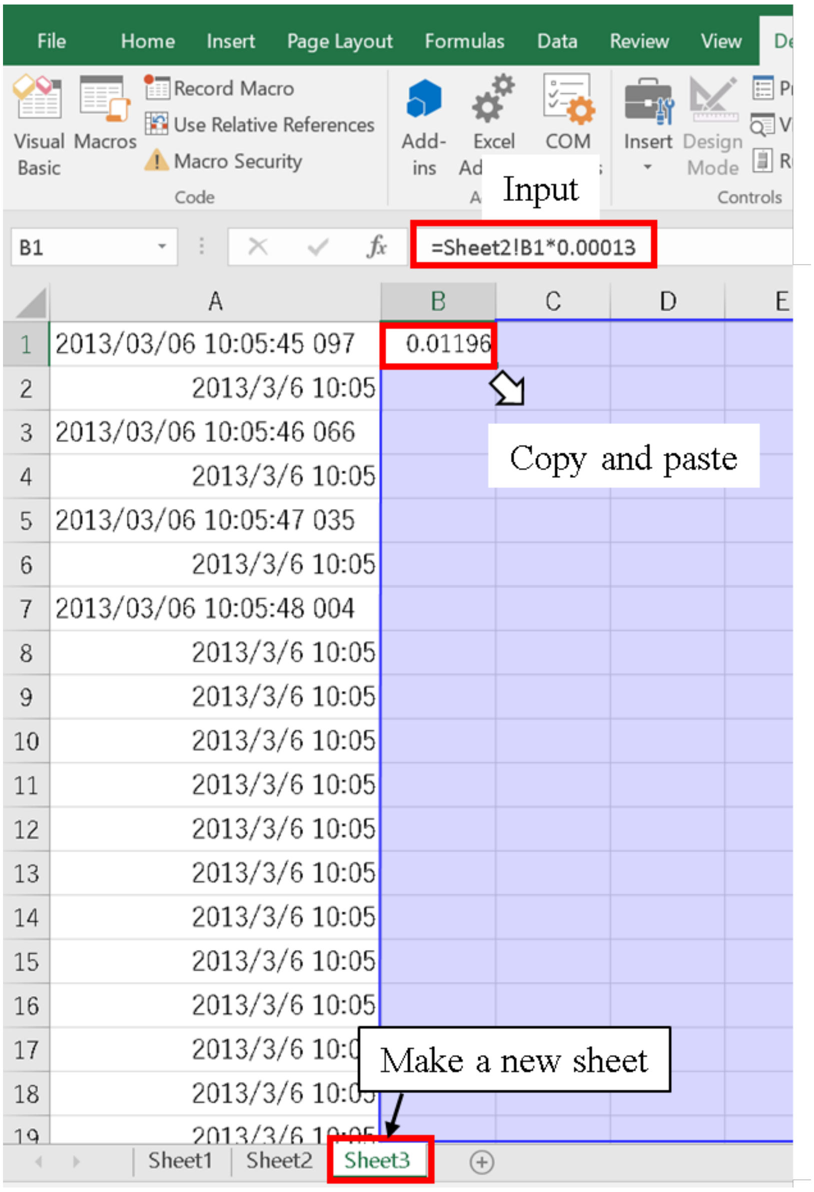


A1 | 2013/03/06 10:05:45 097

	A	B	C	D	E	F
1	2013/03/06 10:05:45 097	[illegible]	29	79	105	10
3	2013/3/6 10:05	-96	4468	9374	11192	9
5	2013/03/06 10:05:46 066	0	1336	6518	9836	10
7	2013/3/6 10	[illegible]	56	6402	10132	10
9	2013/03/06 10:05:47 0	[illegible]	80	9674	11076	10
11	2013/3/6 10:05	68	2832	7946	10432	10
13	2013/03/06 10:05:48 004	92	4696	9514	11516	9
15	2013/3/6 10:05	644	5940	10458	10180	10
17	2013/3/6 10:05	-24	1196	6450	9880	10
19	2013/3/6 10:05	44	4396	9306	11308	9
21	2013/3/6 10:05	-24	2556	7762	10572	10
23	2013/3/6 10:05	68	4972	9838	11216	9
25	2013/3/6 10:05	436	5756	10298	10180	10
27	2013/3/6 10:05	184	5620	10342	10156	10
29	2013/3/6 10:05	68	2232	7394	10296	10
31	2013/3/6 10:05	296	5688	10390	10112	10
33	2013/	[illegible]	04	10206	10180	10
35	2013/	[illegible]	68	8846	10984	10
37	2013/3/6 10:05	68	1220	6450	9904	10

Select destination and press ENTER or choose Paste

Figure 23. Copying and pasting the filtered data to a new sheet.



B1 =Sheet2!B1*0.00013

Input

	A	B	C	D	E
1	2013/03/06 10:05:45 097	0.01196			
2	2013/3/6 10:05				
3	2013/03/06 10:05:46 066				
4	2013/3/6 10:05				
5	2013/03/06 10:05:47 035				
6	2013/3/6 10:05				
7	2013/03/06 10:05:48 004				
8	2013/3/6 10:05				
9	2013/3/6 10:05				
10	2013/3/6 10:05				
11	2013/3/6 10:05				
12	2013/3/6 10:05				
13	2013/3/6 10:05				
14	2013/3/6 10:05				
15	2013/3/6 10:05				
16	2013/3/6 10:05				
17	2013/3/6 10:0				
18	2013/3/6 10:05				
19	2013/3/6 10:05				

Copy and paste

Make a new sheet

Figure 24 Process of converting raw data into voltage data.

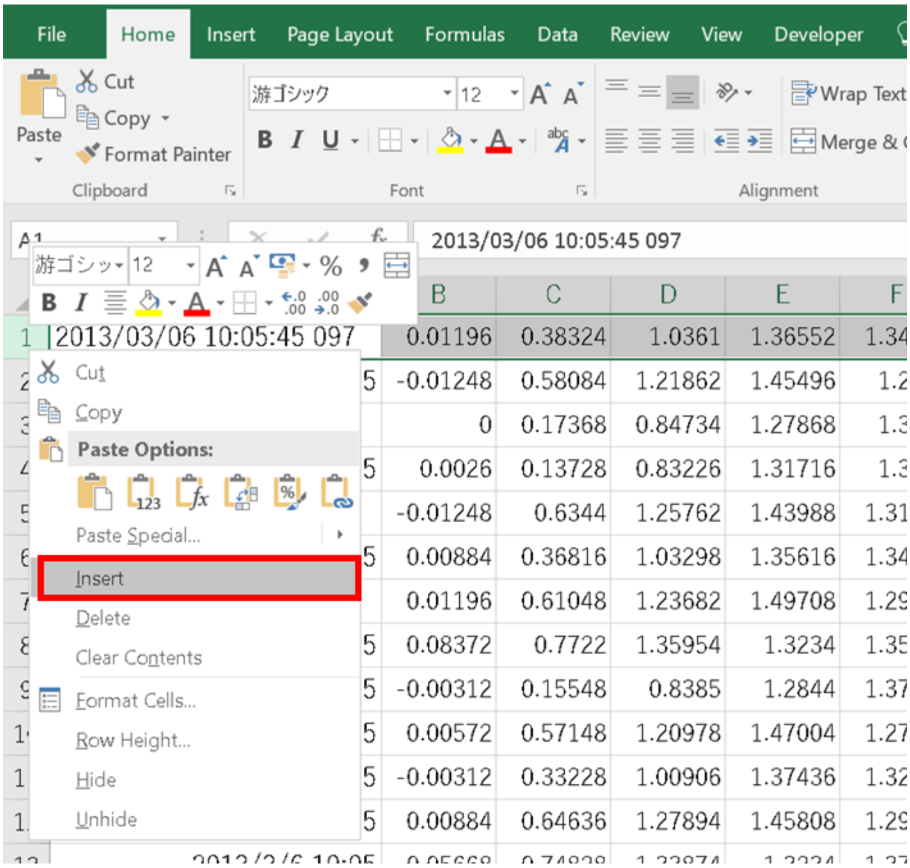
To show the relationship between the voltage data and depth, make a graph as shown in the following processes.

8) Select row 1. Right-click and select “Insert” (Figure 25).

9) Input “Depth” in cell A1 and depth value (meter) in cell B1 to ADU1 (from left cell, 0, 0.0384, 0.0768, …). Insert a “Scatter” graph.

10) Right-click on the graph and select “Select data …”, then the Select Data Source box is opened (Figure 26). Click “Add” in the Select Data Source box. If there is any data series, remove all. And then, Edit Series box is opened. Input Series name, Series x values, and

Series y values. For example, to show the graph of data in row 2, input the data as shown Figure 27, and the graph like in Figure 28 is displayed.



2013/03/06 10:05:45 097

	A	B	C	D	E	F
1	2013/03/06 10:05:45 097	0.01196	0.38324	1.0361	1.36552	1.34
2	5	-0.01248	0.58084	1.21862	1.45496	1.2
3		0	0.17368	0.84734	1.27868	1.3
4	5	0.0026	0.13728	0.83226	1.31716	1.3
5		-0.01248	0.6344	1.25762	1.43988	1.31
6	5	0.00884	0.36816	1.03298	1.35616	1.34
7		0.01196	0.61048	1.23682	1.49708	1.29
8	5	0.08372	0.7722	1.35954	1.3234	1.35
9	5	-0.00312	0.15548	0.8385	1.2844	1.37
10	5	0.00572	0.57148	1.20978	1.47004	1.27
11	5	-0.00312	0.33228	1.00906	1.37436	1.32
12	5	0.00884	0.64636	1.27894	1.45808	1.29

Figure 25. Inserting a new row.

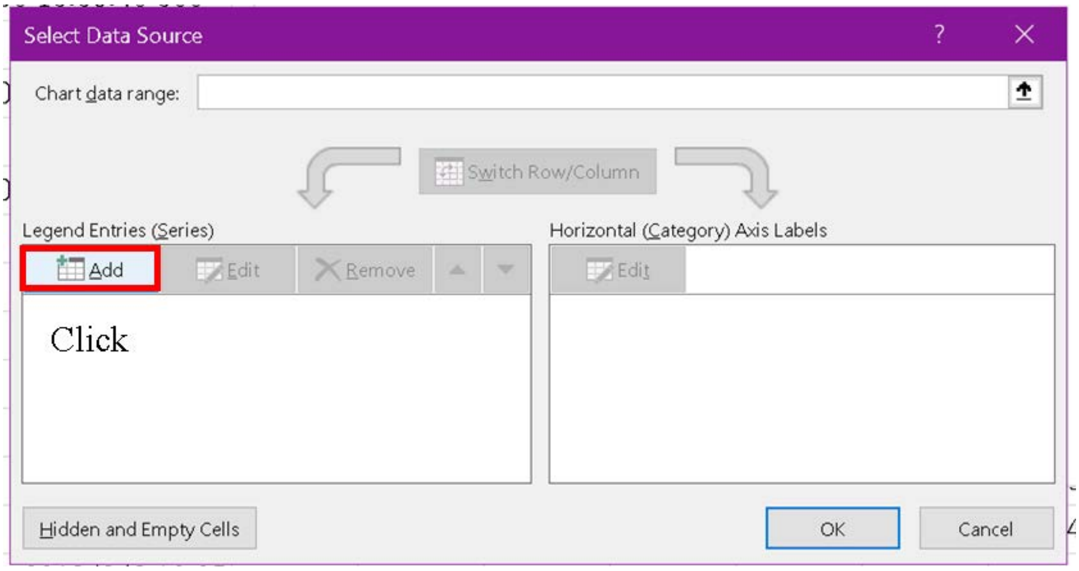
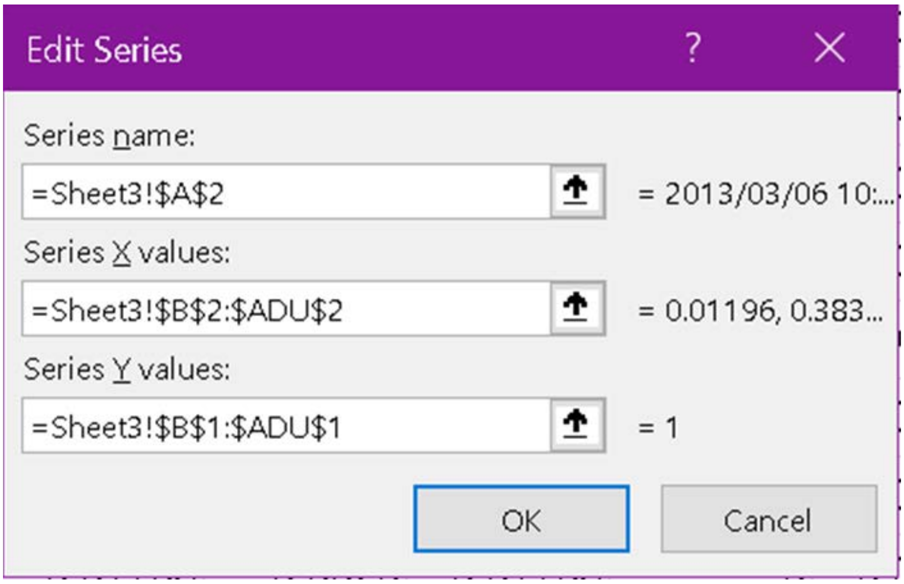


Figure 26. Selecting Data Source.



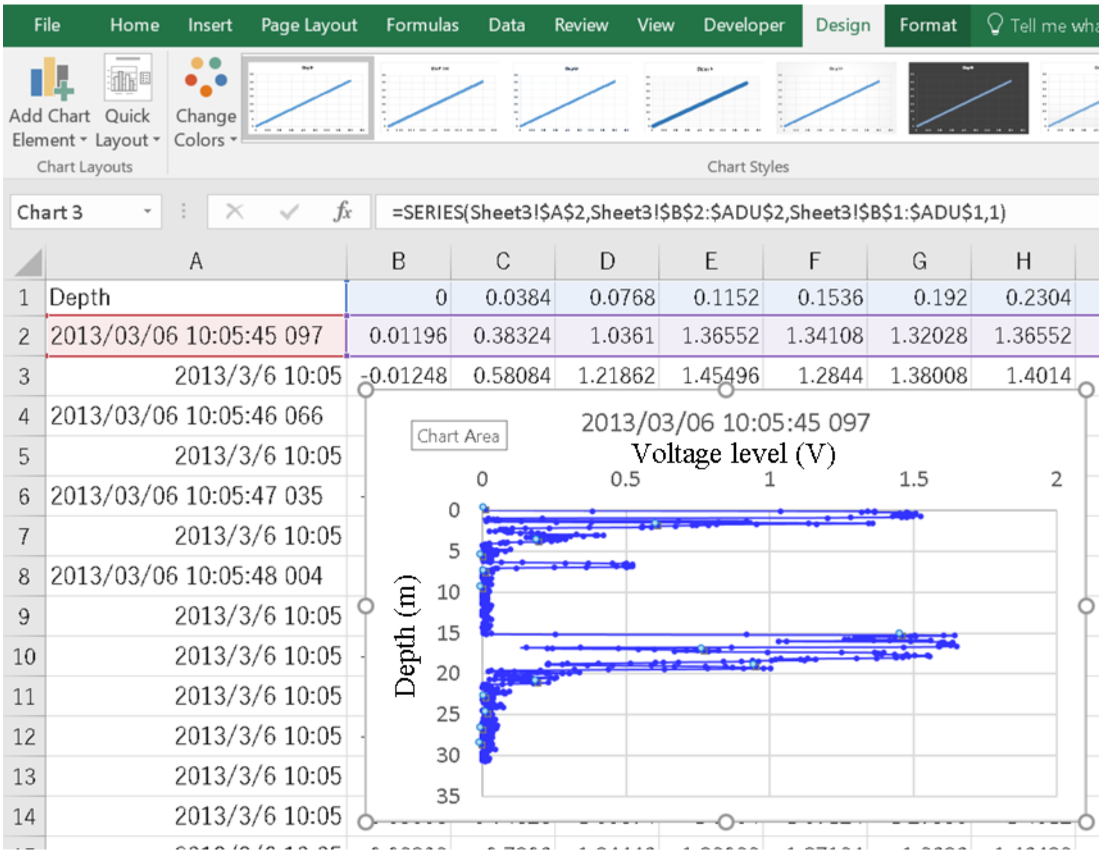
Edit Series

Series name:
=Sheet3!$A$2 = 2013/03/06 10:...

Series X values:
=Sheet3!$B$2:$ADU$2 = 0.01196, 0.383...

Series Y values:
=Sheet3!$B$1:$ADU$1 = 1

Figure 27. Eample of an input data.



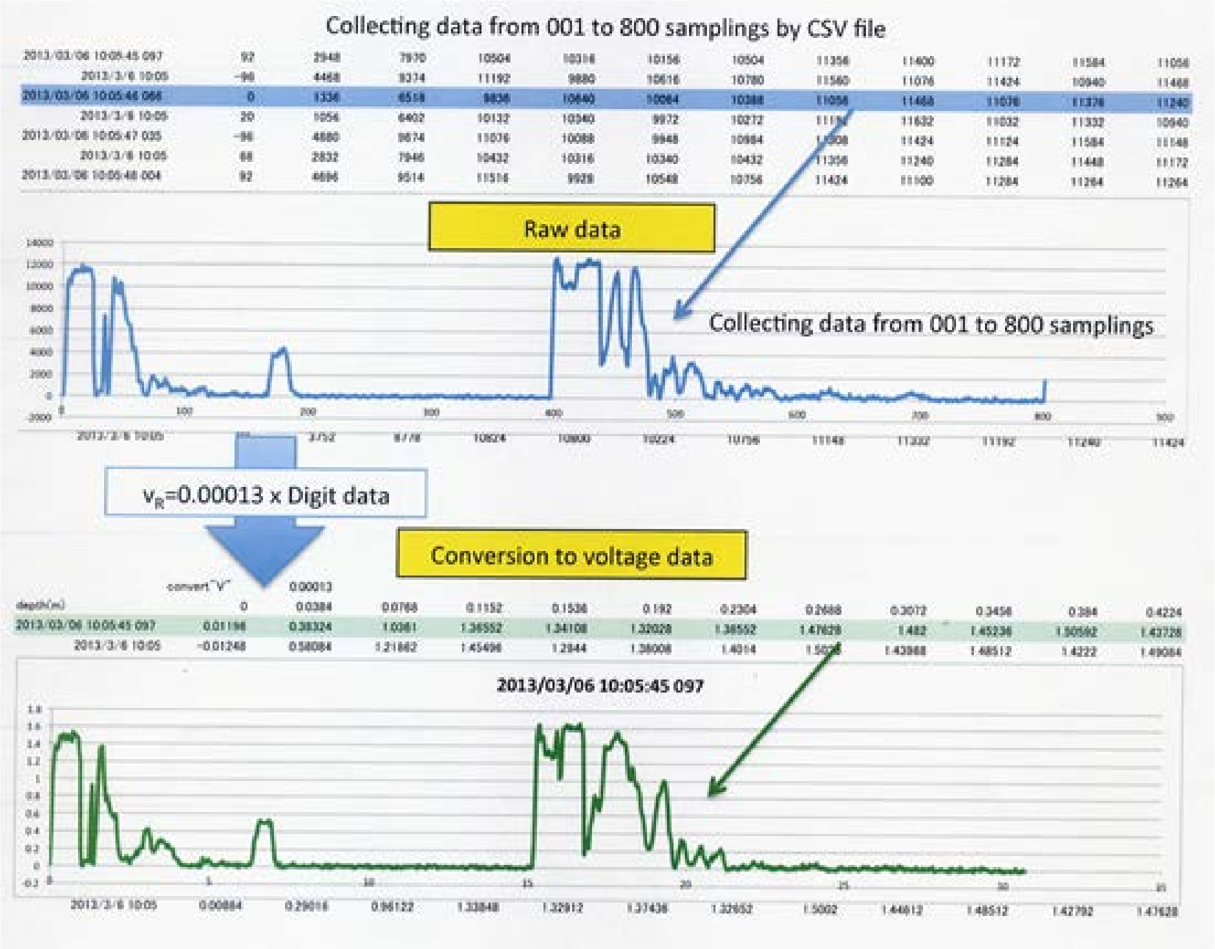
=SERIES(Sheet3!$A$2,Sheet3!$B$2:$ADU$2,Sheet3!$B$1:$ADU$1,1)

	A	B	C	D	E	F	G	H
1	Depth	0	0.0384	0.0768	0.1152	0.1536	0.192	0.2304
2	2013/03/06 10:05:45 097	0.01196	0.38324	1.0361	1.36552	1.34108	1.32028	1.36552
3	2013/3/6 10:05	-0.01248	0.58084	1.21862	1.45496	1.2844	1.38008	1.4014
4	2013/03/06 10:05:46 066							
5	2013/3/6 10:05							
6	2013/03/06 10:05:47 035							
7	2013/3/6 10:05							
8	2013/03/06 10:05:48 004							
9	2013/3/6 10:05							
10	2013/3/6 10:05							
11	2013/3/6 10:05							
12	2013/3/6 10:05							
13	2013/3/6 10:05							
14	2013/3/6 10:05							

Figure 28. Graph showing the relationship between voltage values and depths.

In order to find out the fish abundance distribution in the survey area, a further echo signal data analyzing process is required as shown in the following:

1) Convert digital data (raw data) into voltage data (in terms of linear and decibels), (Figure 29),
2) Calculate the Time Varied Gain (TVG) compensation of loss echo signal for each layer of water depth (Figure 30),
3) Calculate the average value of echo signal for each transmitting ping, (Figure 31),
4) Calculate the integration value of echo signal of each integration distant, (Figure 32),
5) Plot the echo integration value in the survey transect for fish abundance distribution (Figure 33).



Collecting data from 001 to 800 samplings by CSV file

2013/03/06 10:05:45 097	92	2948	7970	10504	10316	10156	10504	11356	11400	11172	11584	11056
2013/3/6 10:05	-96	4468	8374	11192	9880	10616	10780	11560	11076	11424	10940	11468
2013/03/06 10:05:46 066	0	1336	6518	9836	10640	10064	10388	11056	11468	11036	11376	11240
2013/3/6 10:05	20	1056	6402	10132	10340	9972	10272	111[illegible]	11632	11032	11332	10940
2013/03/06 10:05:47 035	-96	4880	9674	11076	10088	9948	10984	1[illegible]308	11424	11124	11584	11148
2013/3/6 10:05	68	2832	7946	10432	10316	10340	10432	11356	11240	11284	11448	11172
2013/03/06 10:05:48 004	92	4696	9514	11516	9928	10548	10756	11424	11100	11284	11264	11264

Raw data

Collecting data from 001 to 800 samplings

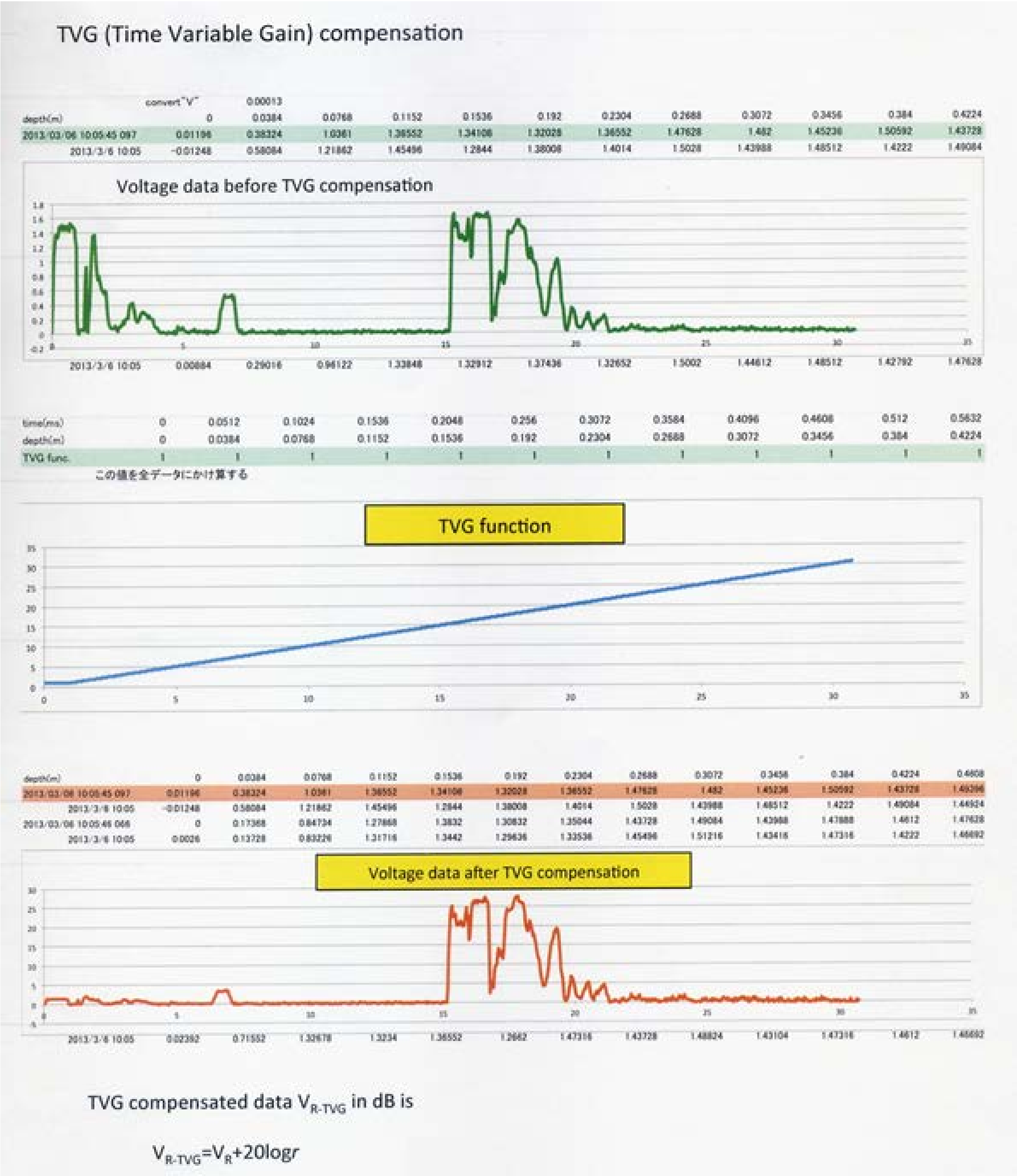
V_R=0.00013 x Digit data

Conversion to voltage data

convert"V"		0.00013										
depth(m)	0	0.0384	0.0768	0.1152	0.1536	0.192	0.2304	0.2688	0.3072	0.3456	0.384	0.4224
2013/03/06 10:05:45 097	0.01196	0.38324	1.0361	1.36552	1.34108	1.32028	1.38552	1.47628	1.482	1.45236	1.50592	1.43728
2013/3/6 10:05	-0.01248	0.58084	1.21862	1.45496	1.2844	1.38008	1.4014	1.50[illegible]	1.43988	1.48512	1.4222	1.49084

2013/03/06 10:05:45 097

Figure 29. Converting digital data (raw data) in to voltage data.



TVG (Time Variable Gain) compensation

	convert "V"	0.00013										
depth(m)	0	0.0384	0.0768	0.1152	0.1536	0.192	0.2304	0.2688	0.3072	0.3456	0.384	0.4224
2013/03/06 10:05:45 097	0.01196	0.38324	1.0361	1.36552	1.34108	1.32028	1.36552	1.47628	1.482	1.45236	1.50592	1.43728
2013/3/6 10:05	-0.01248	0.58084	1.21862	1.45496	1.2844	1.38008	1.4014	1.5028	1.43988	1.48512	1.4222	1.49084

Voltage data before TVG compensation

2013/3/6 10:05	0.00884	0.29016	0.96122	1.33848	1.32912	1.37436	1.32652	1.5002	1.44612	1.48512	1.42792	1.47628

time(ms)	0	0.0512	0.1024	0.1536	0.2048	0.256	0.3072	0.3584	0.4096	0.4608	0.512	0.5632
depth(m)	0	0.0384	0.0768	0.1152	0.1536	0.192	0.2304	0.2688	0.3072	0.3456	0.384	0.4224
TVG func.	1	1	1	1	1	1	1	1	1	1	1	1

この値を全データにかけ算する

TVG function

depth(m)	0	0.0384	0.0768	0.1152	0.1536	0.192	0.2304	0.2688	0.3072	0.3456	0.384	0.4224	0.4608
2013/03/06 10:05:45 097	0.01196	0.38324	1.0361	1.36552	1.34108	1.32028	1.36552	1.47628	1.482	1.45236	1.50592	1.43728	1.49396
2013/3/6 10:05	-0.01248	0.58084	1.21862	1.45496	1.2844	1.38008	1.4014	1.5028	1.43988	1.48512	1.4222	1.49084	1.46924
2013/03/06 10:05:46 066	0	0.17368	0.84734	1.27868	1.3832	1.30832	1.35044	1.43728	1.49084	1.43988	1.47888	1.4612	1.47628
2013/3/6 10:05	0.0026	0.13728	0.83226	1.31716	1.3442	1.29636	1.33536	1.45496	1.51216	1.43416	1.47316	1.4222	1.46692

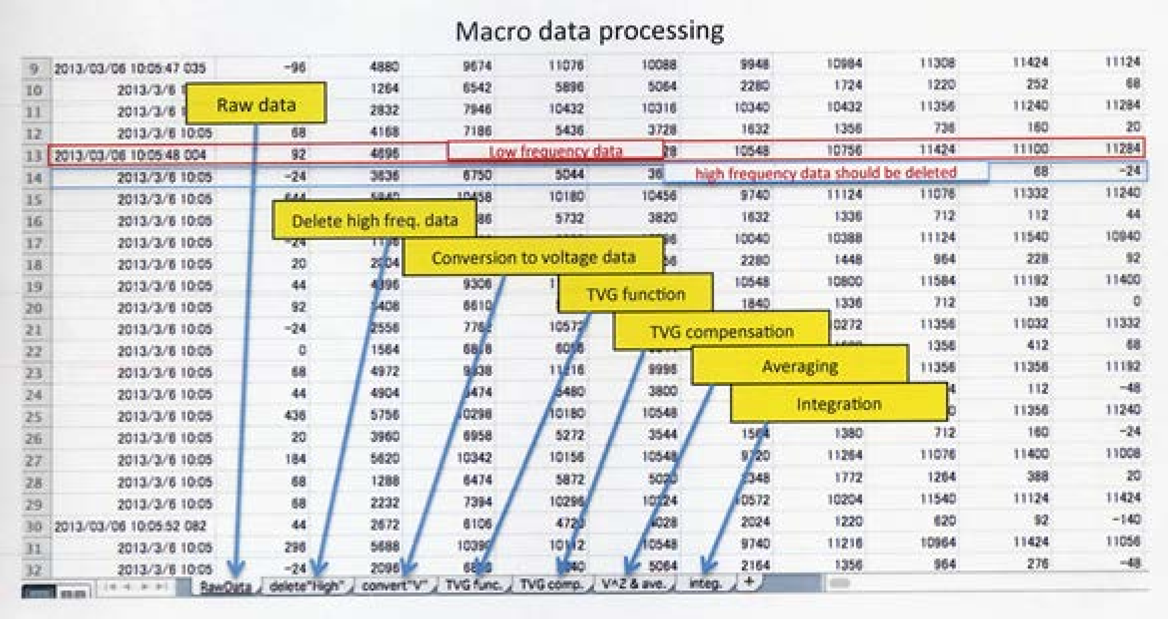
Voltage data after TVG compensation

2013/3/6 10:05	0.02392	0.71552	1.32678	1.3234	1.36552	1.2662	1.47316	1.43728	1.48824	1.43104	1.47316	1.4612	1.46692

TVG compensated data $V_{R\text{-}TVG}$ in dB is

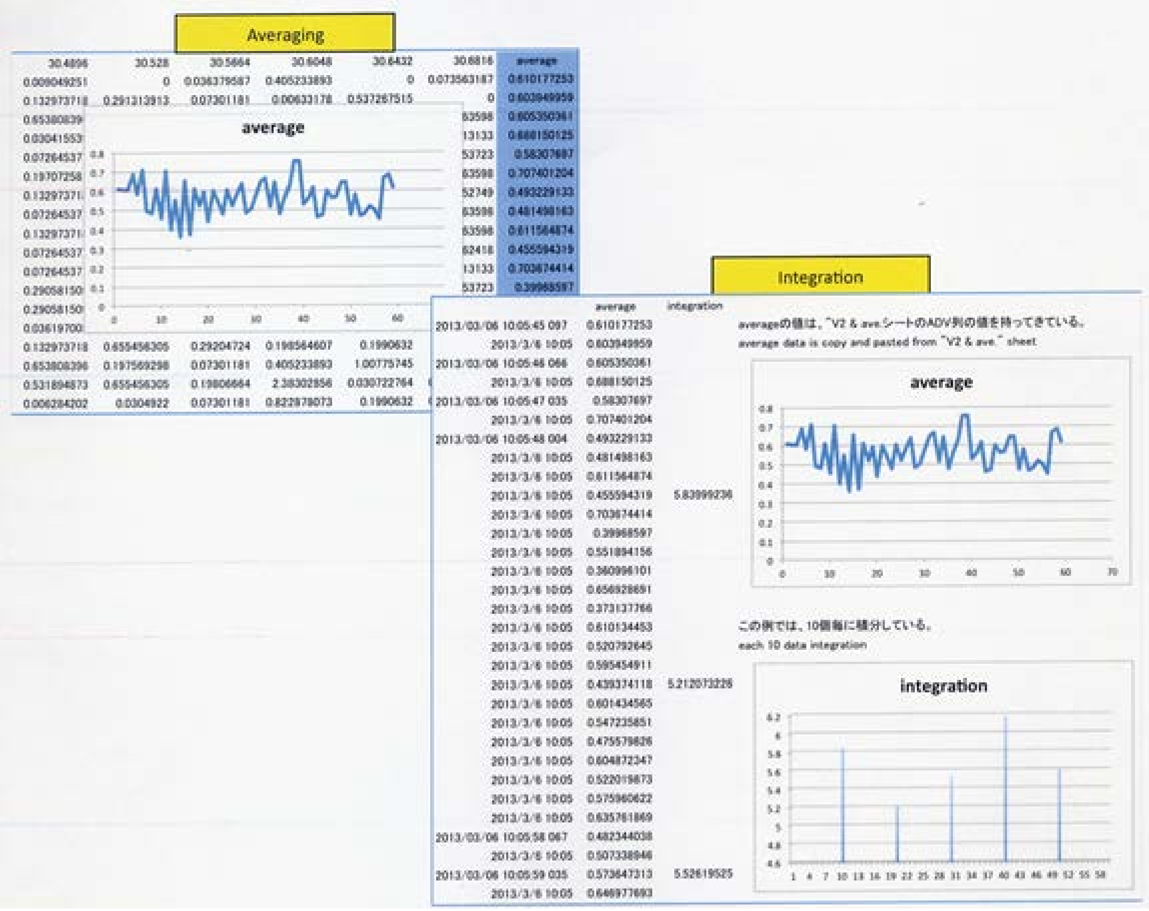
$$V_{R\text{-}TVG}=V_R+20\log r$$

Figure 30. Calculation of Time Varied Gain (TVG) compensation of loss echo signal for each layer of water depth.



9	2013/03/06 10:05:47 035	-96	4880	9674	11076	10088	9948	10984	11308	11424	11124
10	2013/3/6 [illegible]	[illegible]	1264	6542	5896	5064	2280	1724	1220	252	68
11	2013/3/6 [illegible]	[illegible]	2832	7946	10432	10316	10340	10432	11356	11240	11284
12	2013/3/6 10:05	68	4168	7186	5436	3728	1632	1356	736	160	20
13	2013/03/06 10:05:48 004	92	4696	[illegible]	[illegible]	[illegible]	10548	10756	11424	11100	11284
14	2013/3/6 10:05	-24	3636	6750	5044	[illegible]	[illegible]	[illegible]	[illegible]	68	-24
15	2013/3/6 10:05	[illegible]	[illegible]	[illegible]	10180	10456	9740	11124	11076	11332	11240
16	2013/3/6 10:05	[illegible]	[illegible]	[illegible]	5732	3820	1632	1336	712	112	44
17	2013/3/6 10:05	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	10040	10388	11124	11540	10940
18	2013/3/6 10:05	20	[illegible]	[illegible]	[illegible]	[illegible]	2280	1448	964	228	92
19	2013/3/6 10:05	44	[illegible]	9306	[illegible]	[illegible]	10548	10800	11584	11192	11400
20	2013/3/6 10:05	92	[illegible]	6610	[illegible]	[illegible]	1840	1336	712	136	0
21	2013/3/6 10:05	-24	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	11356	11032	11332
22	2013/3/6 10:05	0	1564	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1356	412	68
23	2013/3/6 10:05	68	4972	[illegible]	[illegible]	9996	[illegible]	[illegible]	11356	11356	11192
24	2013/3/6 10:05	44	4904	[illegible]	[illegible]	3800	[illegible]	[illegible]	[illegible]	112	-48
25	2013/3/6 10:05	436	5756	[illegible]	[illegible]	10548	[illegible]	[illegible]	[illegible]	11356	11240
26	2013/3/6 10:05	20	3960	6958	5272	3544	[illegible]	1380	712	160	-24
27	2013/3/6 10:05	184	5620	10342	10156	[illegible]	[illegible]	11264	11076	11400	11008
28	2013/3/6 10:05	68	1288	6474	5872	[illegible]	[illegible]	1772	1264	388	20
29	2013/3/6 10:05	68	2232	7394	[illegible]	[illegible]	[illegible]	10204	11540	11124	11424
30	2013/03/06 10:05:52 082	44	2672	6106	[illegible]	[illegible]	2024	1220	620	92	-140
31	2013/3/6 10:05	296	5688	[illegible]	[illegible]	[illegible]	9740	11216	10964	11424	11056
32	2013/3/6 10:05	-24	2096	[illegible]	[illegible]	5064	2164	1356	964	276	-48

Figure 31. Calculation of average value of echo signal for each transmitting ping.



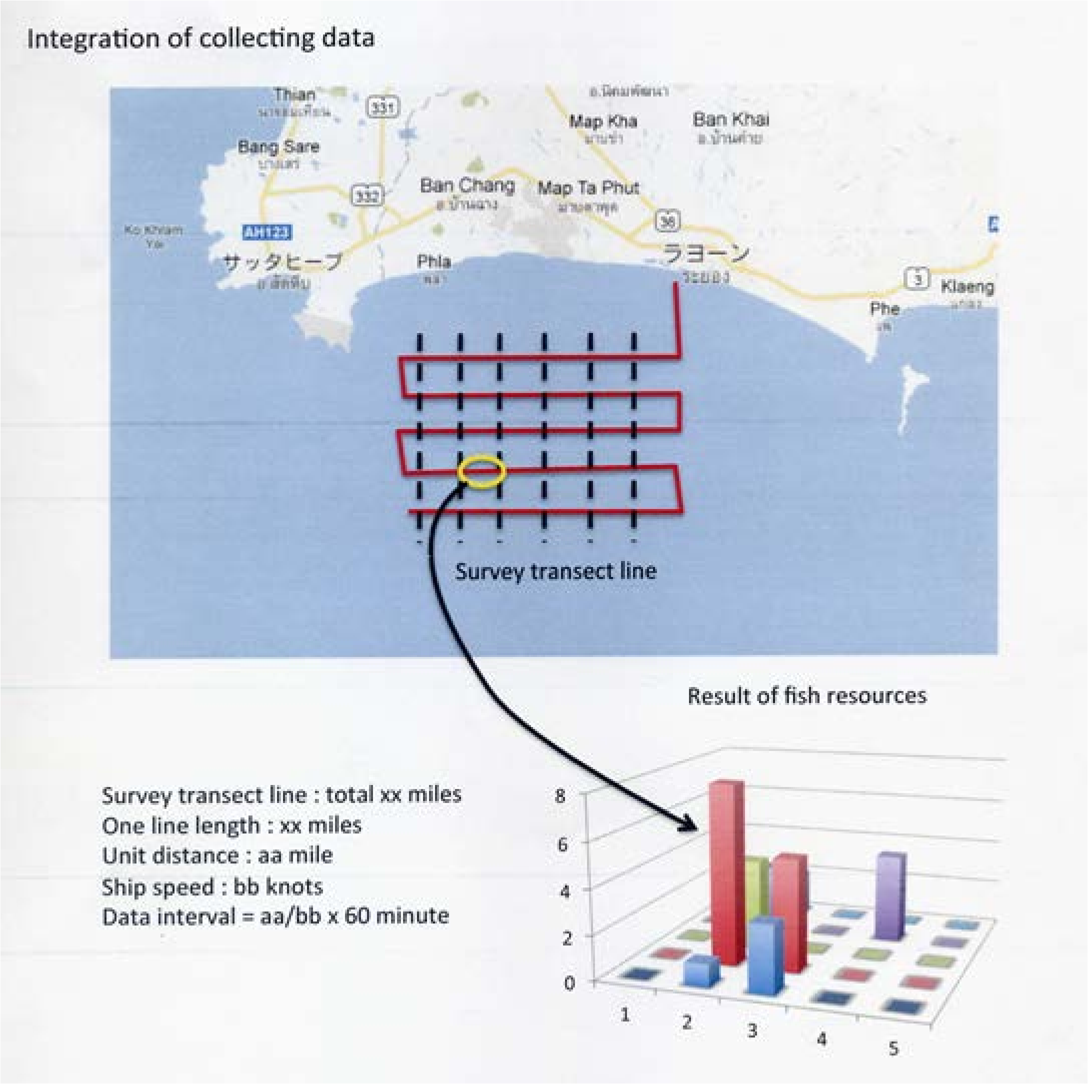
30.4896	30.528	30.5664	30.6048	30.6432	30.6816	average
0.009049251	0	0.036379587	0.405233893	0	0.073563187	0.610177253
0.132973718	0.291313913	0.07301181	0.00633178	0.537267515	0	0.603949959
0.65380839[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]63598	0.605350361
0.03041553[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]13133	0.688150125
0.07264537[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]53723	0.58307697
0.19707258[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]63598	0.707401204
0.13297371[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]52749	0.493229133
0.07264537[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]63598	0.481498163
0.13297371[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]63598	0.611564874
0.07264537[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]62418	0.455594319
0.07264537[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]13133	0.703674414
0.29058150[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]53723	0.39968597
0.29058150[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
0.03619700[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
0.132973718	0.655456305	0.29204724	0.198564607	0.1990632	[illegible]	[illegible]
0.653808396	0.197569298	0.07301181	0.405233893	1.00775745	[illegible]	[illegible]
0.521894873	0.655456305	0.19806664	2.38302856	0.030722764	[illegible]	[illegible]
0.006284202	0.0304922	0.07301181	0.822878073	0.1990632	[illegible]	[illegible]

	average	integration
2013/03/06 10:05:45 097	0.610177253	
2013/3/6 10:05	0.603949959	
2013/03/06 10:05:46 066	0.605350361	
2013/3/6 10:05	0.688150125	
2013/03/06 10:05:47 035	0.58307697	
2013/3/6 10:05	0.707401204	
2013/03/06 10:05:48 004	0.493229133	
2013/3/6 10:05	0.481498163	
2013/3/6 10:05	0.611564874	
2013/3/6 10:05	0.455594319	5.83999236
2013/3/6 10:05	0.703674414	
2013/3/6 10:05	0.39968597	
2013/3/6 10:05	0.551894156	
2013/3/6 10:05	0.360996101	
2013/3/6 10:05	0.656928691	
2013/3/6 10:05	0.373137766	
2013/3/6 10:05	0.610134453	
2013/3/6 10:05	0.520792645	
2013/3/6 10:05	0.595454911	
2013/3/6 10:05	0.439374118	5.212073226
2013/3/6 10:05	0.601434565	
2013/3/6 10:05	0.547235851	
2013/3/6 10:05	0.475579826	
2013/3/6 10:05	0.604872347	
2013/3/6 10:05	0.522019873	
2013/3/6 10:05	0.575960622	
2013/3/6 10:05	0.635761869	
2013/03/06 10:05:58 067	0.482344038	
2013/3/6 10:05	0.507338946	
2013/03/06 10:05:59 035	0.573647313	5.52619525
2013/3/6 10:05	0.646977693	

averageの値は、"V2 & ave.シートのADV列の値を持ってきている。
average data is copy and pasted from "V2 & ave." sheet

この例では、10個毎に積分している。
each 10 data integration

Figure 32. Calculation of integration value of echo signal of each integration distance.


Integration of collecting data

Survey transect line : total xx miles
One line length : xx miles
Unit distance : aa mile
Ship speed : bb knots
Data interval = aa/bb x 60 minute

Figure 33. Plot of the echo integration value in the survey transect for fish abundance distribution**.**

GPS DATA ANALYSIS PROCEDURE

Outline of GPS data

The GPS receiver output data or record tracks, the time, accuracy, etc. in NMEA format. NMEA is acronym for the National Marine Electronics Association. There are different types of NMEA messages. Some of those that are applicable to GPS receivers are listed below.

- GPGGA: essential fix data which provides 3D location and accuracy data.
- GPRMC: has its own version of essential GPS pvt (position, velocity, time) data.
- GPDPT: Depth
- GPVTG: Speed over ground and tracking offset.

The GPGGA message has enough information for this Manual's system. To understand the NMEA message structure, examine the $GPGGA message as shown below, this particular message was an output from GPS receiver:

```
$GPGGA,031959,1235.0071,N,10122.5079,E,1,16,0.7,3.6,M,,,,*0C
```

031959 is the time stamp: UTC time in hours, minutes and seconds.

1235.0071 is the latitude in the DDMM.MMMM format.

N denotes north latitude.

10122.5079 is the longitude in the DDDMM.MMMM format.

E denotes east longitude.

1 denotes the fixed quality: 1 = independent
2 = Differentially correct coordinate (e.g., WAAS, DGPS)
3 = PPS fix
4 = RTK fix coordinate (centimeter precision)
5 = RTK Float (decimeter precision.

16 denotes number of satellites used in the coordinate.

0.7 denotes the HDOP (horizontal dilution of precision).

3.6 denotes altitude, meters, above mean sea level.

M denotes units of altitude (eg. meters or feet)

(empty field): Height of geoid (mean sea level) above WGS84 ellipsoid

(empty field) time in seconds since last DGPS update

(empty field) DGPS station ID number

*0c is the checksum data, always begins with *

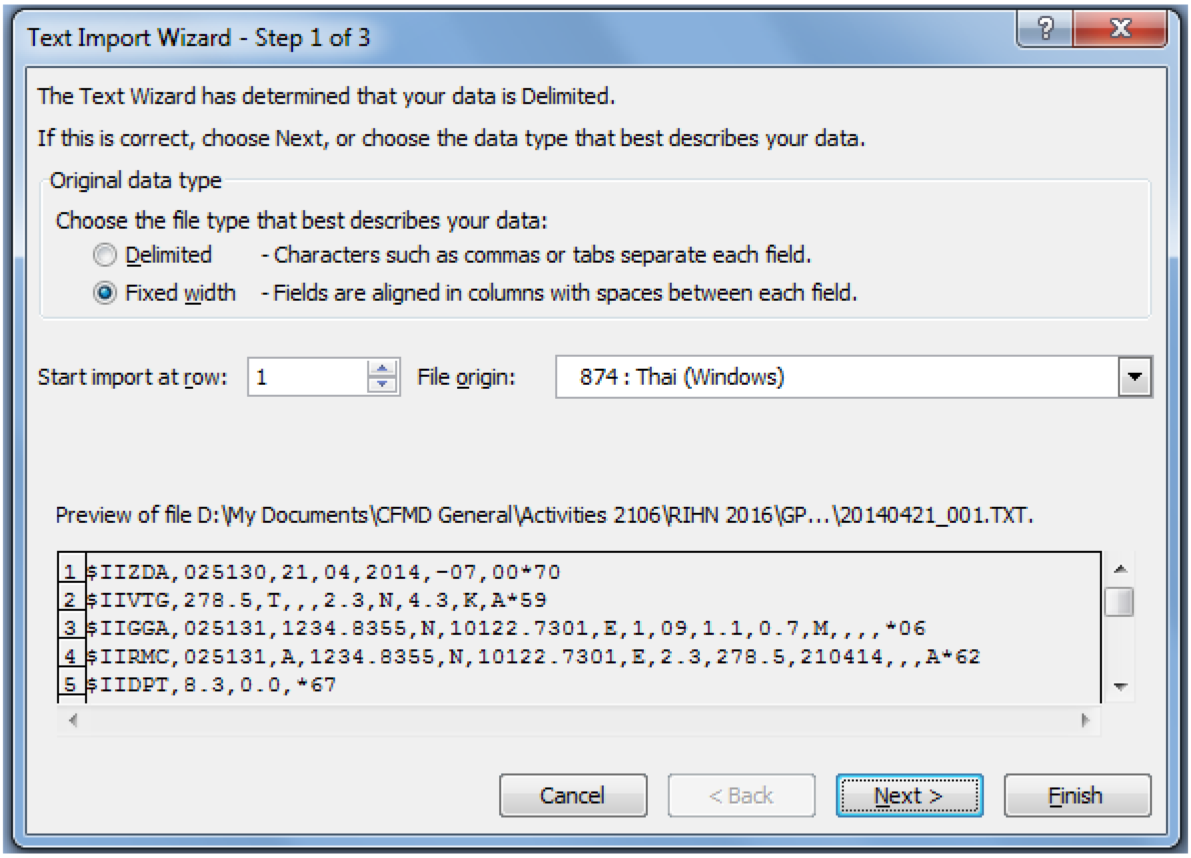
The NMEA of GPS is saved in text format. Here, the resource distribution is mapped with Microsoft Excel.

a. Loading NMEA data with Excel

Open text file data by using Excel program

File -> Open --> All Files

At Text Import Wizard – Step 1 of 3, select "Fix width", "Next"



Text Import Wizard - Step 1 of 3

The Text Wizard has determined that your data is Delimited.

If this is correct, choose Next, or choose the data type that best describes your data.

Original data type

Choose the file type that best describes your data:

○ Delimited - Characters such as commas or tabs separate each field.

● Fixed width - Fields are aligned in columns with spaces between each field.

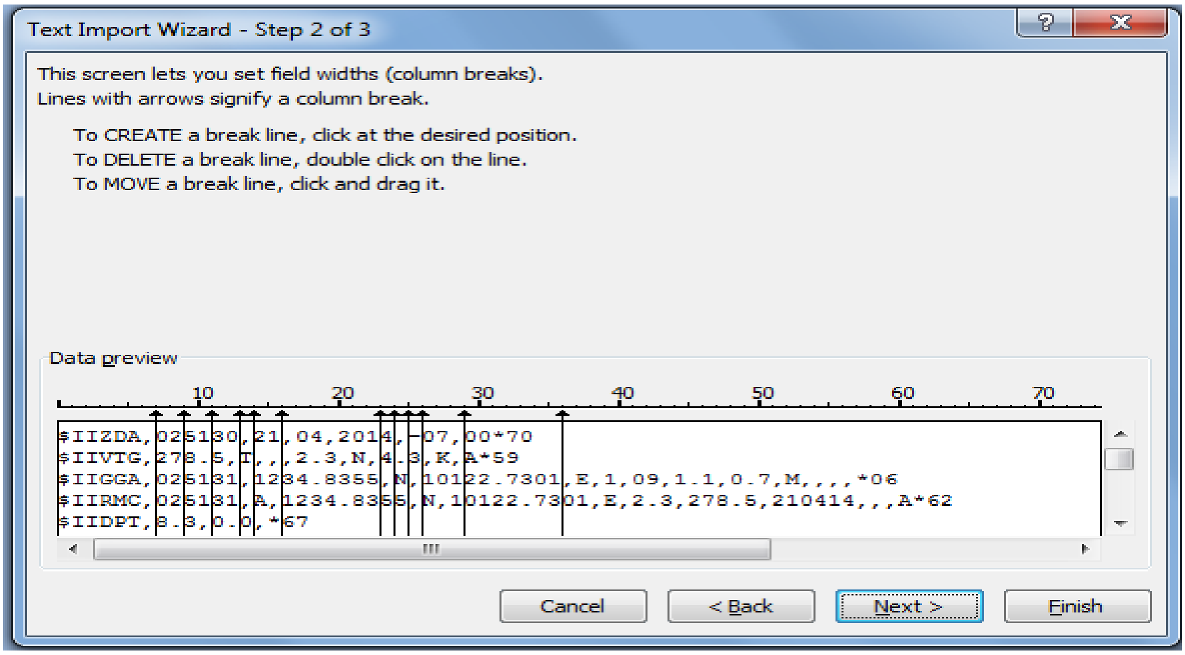
Start import at row: 1

File origin: 874 : Thai (Windows)

Preview of file D:\My Documents\CFMD General\Activities 2106\RIHN 2016\GP...\20140421_001.TXT.

```
$IIZDA,025130,21,04,2014,-07,00*70
$IIVTG,278.5,T,,,2.3,N,4.3,K,A*59
$IIGGA,025131,1234.8355,N,10122.7301,E,1,09,1.1,0.7,M,,,,*06
$IIRMC,025131,A,1234.8355,N,10122.7301,E,2.3,278.5,210414,,,A*62
$IIDPT,8.3,0.0,*67
```

b. At Text Import Wizard – Step 2 of 3, separate cells by hour, minute, second, degree, min., lat., degree, minute, and long., and select “Next”



Text Import Wizard - Step 2 of 3

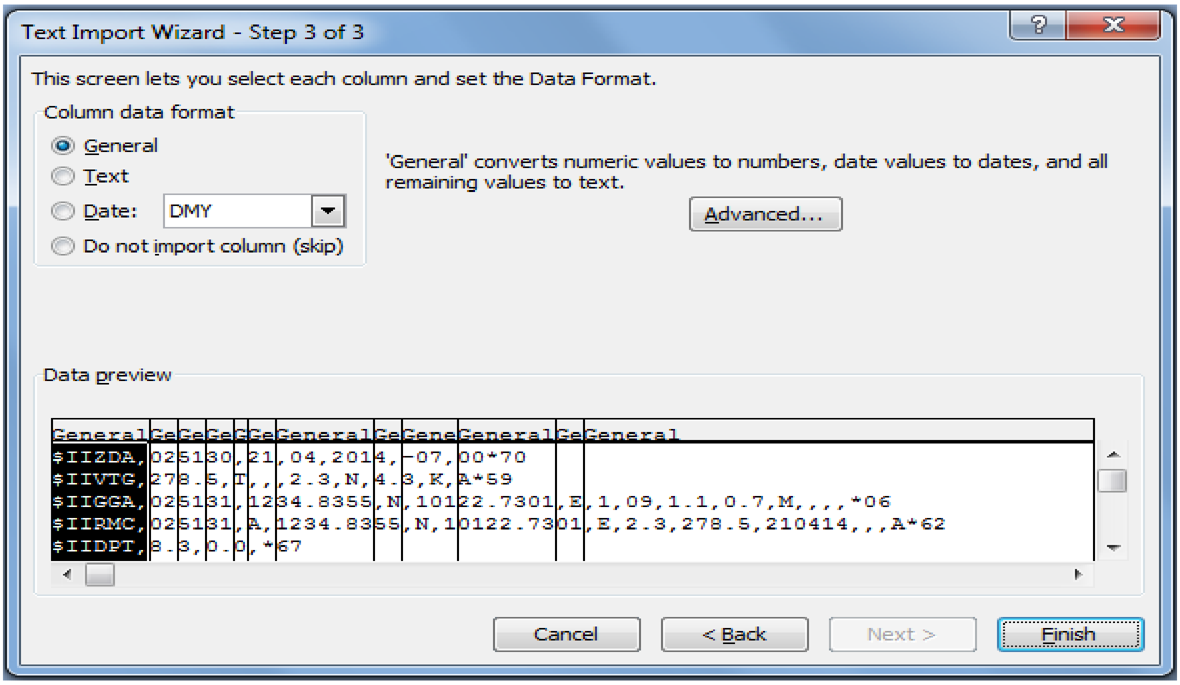
This screen lets you set field widths (column breaks).
Lines with arrows signify a column break.

To CREATE a break line, click at the desired position.
To DELETE a break line, double click on the line.
To MOVE a break line, click and drag it.

Data preview

```
$IIZDA,025130,21,04,2014,-07,00*70
$IIVTG,278.5,T,,,2.3,N,4.3,K,A*59
$IIGGA,025131,1234.8355,N,10122.7301,E,1,09,1.1,0.7,M,,,,*06
$IIRMC,025131,A,1234.8355,N,10122.7301,E,2.3,278.5,210414,,,A*62
$IIDPT,8.3,0.0,*67
```

c. At Text Import Wizard – Step 3 of 3, select “General”, “Finish”



Text Import Wizard - Step 3 of 3

This screen lets you select each column and set the Data Format.

Column data format

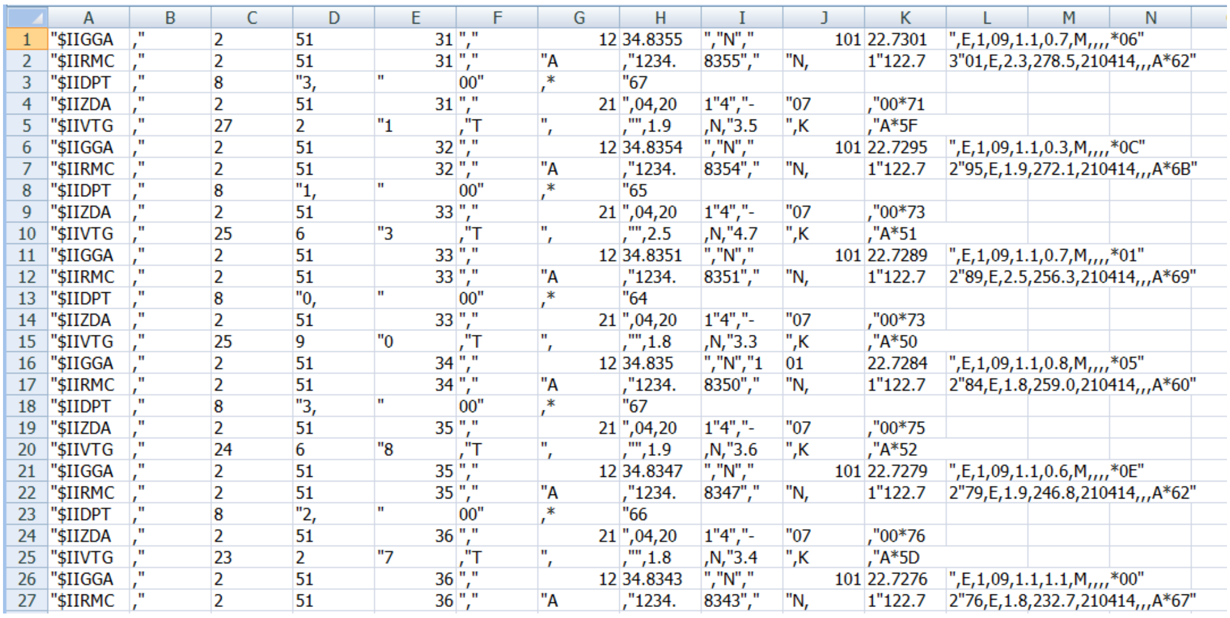
● General
○ Text
○ Date: DMY
○ Do not import column (skip)

'General' converts numeric values to numbers, date values to dates, and all remaining values to text.

Data preview

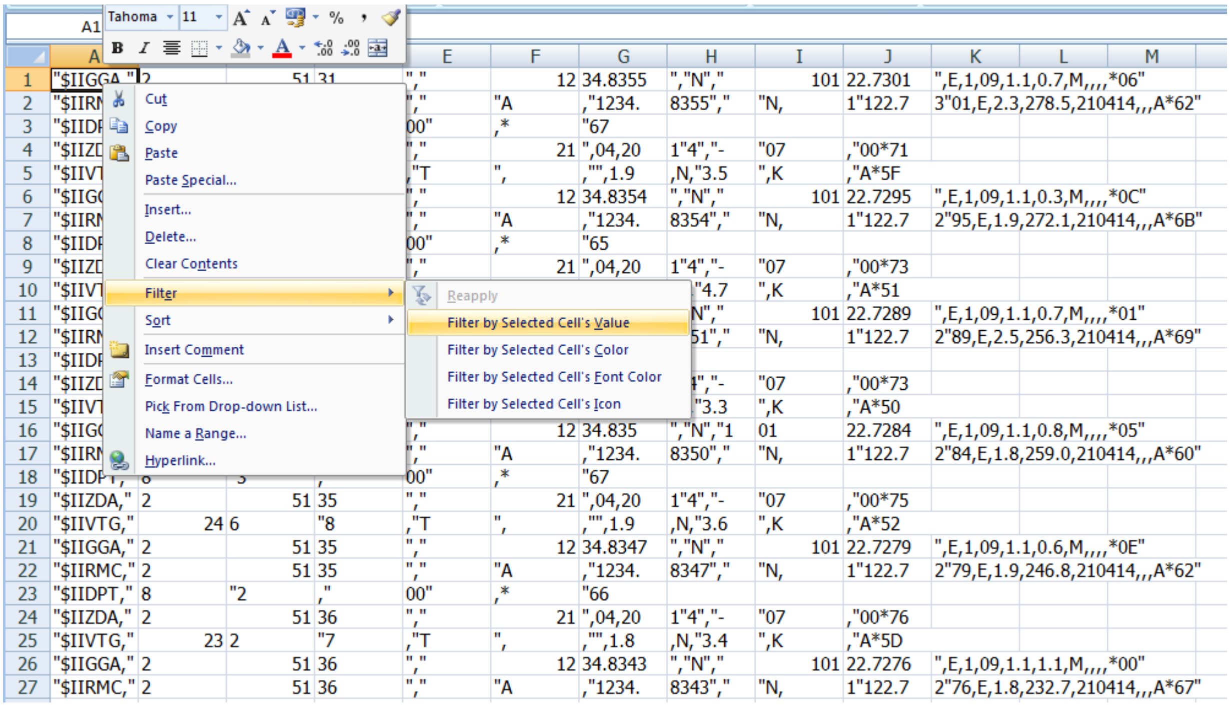
```
GeneralGeGeGeGGeGeneralGeGeneGeneralGeGeneral
$IIZDA,025130,21,04,2014,-07,00*70
$IIVTG,278.5,T,,,2.3,N,4.3,K,A*59
$IIGGA,025131,1234.8355,N,10122.7301,E,1,09,1.1,0.7,M,,,,*06
$IIRMC,025131,A,1234.8355,N,10122.7301,E,2.3,278.5,210414,,,A*62
$IIDPT,8.3,0.0,*67
```

d. The Excel data sheet will be displayed as follows:

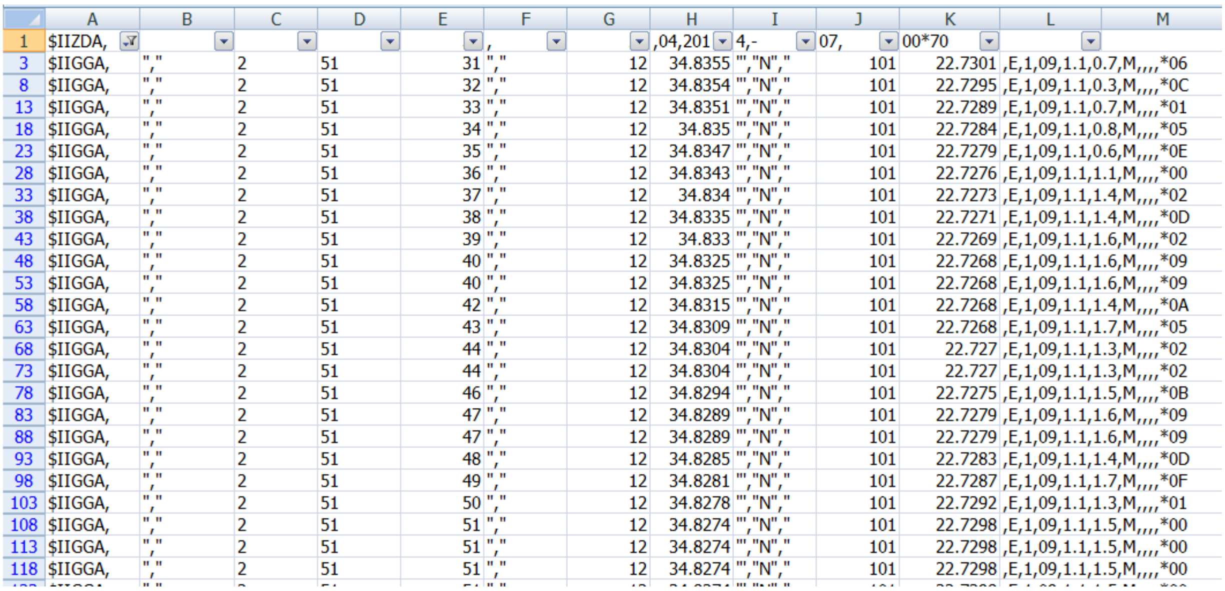


	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	"$IIGGA	,"	2	51	31	","	12	34.8355	","N","	101	22.7301	",E,1,09,1.1,0.7,M,,,,*06"		
2	"$IIRMC	,"	2	51	31	","	"A	,"1234.	8355","	"N,	1"122.7	3"01,E,2.3,278.5,210414,,,A*62"		
3	"$IIDPT	,"	8	"3,	"	00"	,*	"67						
4	"$IIZDA	,"	2	51	31	","	21	",04,20	1"4","-	"07	,"00*71			
5	"$IIVTG	,"	27	2	"1	,"T	",	,"",1.9	,N,"3.5	",K	,"A*5F			
6	"$IIGGA	,"	2	51	32	","	12	34.8354	","N","	101	22.7295	",E,1,09,1.1,0.3,M,,,,*0C"		
7	"$IIRMC	,"	2	51	32	","	"A	,"1234.	8354","	"N,	1"122.7	2"95,E,1.9,272.1,210414,,,A*6B"		
8	"$IIDPT	,"	8	"1,	"	00"	,*	"65						
9	"$IIZDA	,"	2	51	33	","	21	",04,20	1"4","-	"07	,"00*73			
10	"$IIVTG	,"	25	6	"3	,"T	",	,"",2.5	,N,"4.7	",K	,"A*51			
11	"$IIGGA	,"	2	51	33	","	12	34.8351	","N","	101	22.7289	",E,1,09,1.1,0.7,M,,,,*01"		
12	"$IIRMC	,"	2	51	33	","	"A	,"1234.	8351","	"N,	1"122.7	2"89,E,2.5,256.3,210414,,,A*69"		
13	"$IIDPT	,"	8	"0,	"	00"	,*	"64						
14	"$IIZDA	,"	2	51	33	","	21	",04,20	1"4","-	"07	,"00*73			
15	"$IIVTG	,"	25	9	"0	,"T	",	,"",1.8	,N,"3.3	",K	,"A*50			
16	"$IIGGA	,"	2	51	34	","	12	34.835	","N","1	01	22.7284	",E,1,09,1.1,0.8,M,,,,*05"		
17	"$IIRMC	,"	2	51	34	","	"A	,"1234.	8350","	"N,	1"122.7	2"84,E,1.8,259.0,210414,,,A*60"		
18	"$IIDPT	,"	8	"3,	"	00"	,*	"67						
19	"$IIZDA	,"	2	51	35	","	21	",04,20	1"4","-	"07	,"00*75			
20	"$IIVTG	,"	24	6	"8	,"T	",	,"",1.9	,N,"3.6	",K	,"A*52			
21	"$IIGGA	,"	2	51	35	","	12	34.8347	","N","	101	22.7279	",E,1,09,1.1,0.6,M,,,,*0E"		
22	"$IIRMC	,"	2	51	35	","	"A	,"1234.	8347","	"N,	1"122.7	2"79,E,1.9,246.8,210414,,,A*62"		
23	"$IIDPT	,"	8	"2,	"	00"	,*	"66						
24	"$IIZDA	,"	2	51	36	","	21	",04,20	1"4","-	"07	,"00*76			
25	"$IIVTG	,"	23	2	"7	,"T	",	,"",1.8	,N,"3.4	",K	,"A*5D			
26	"$IIGGA	,"	2	51	36	","	12	34.8343	","N","	101	22.7276	",E,1,09,1.1,1.1,M,,,,*00"		
27	"$IIRMC	,"	2	51	36	","	"A	,"1234.	8343","	"N,	1"122.7	2"76,E,1.8,232.7,210414,,,A*67"		

e. Move cursor into cell which indicates “$IIGGA” where information on time, and position (Latitude and Longitude) are contained, use right bottom mouse click for drop down menu. Select “Filter”, “Filter by Selected Cell’s Value”.


Tahoma
11
Cut
Copy
Paste
Paste Special...
Insert...
Delete...
Clear Contents
Filter
Sort
Insert Comment
Format Cells...
Pick From Drop-down List...
Name a Range...
Hyperlink...
Reapply
Filter by Selected Cell's Value
Filter by Selected Cell's Color
Filter by Selected Cell's Font Color
Filter by Selected Cell's Icon

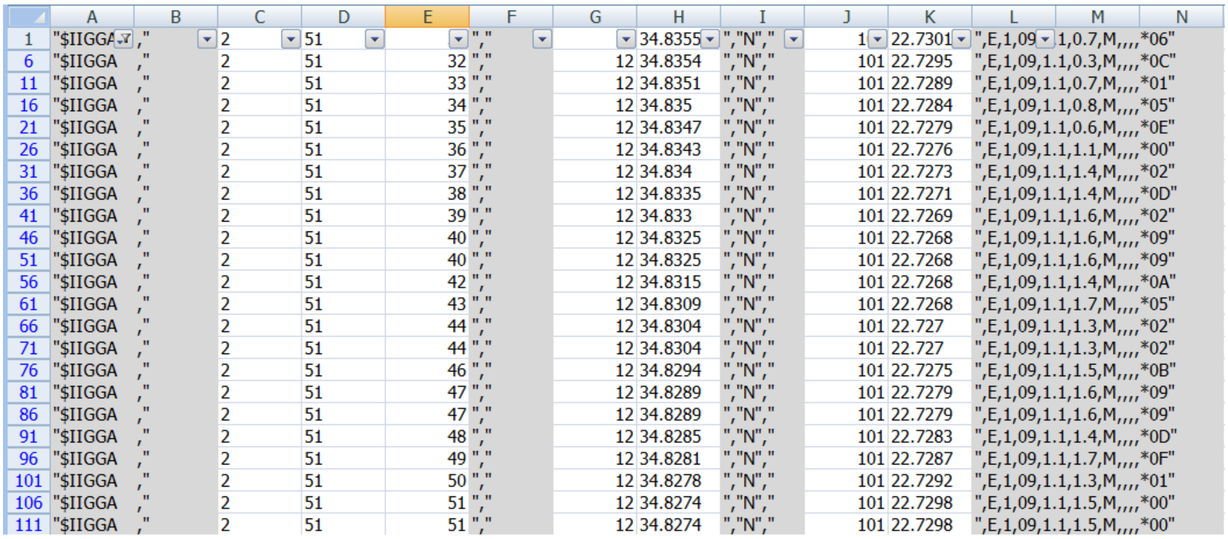
f. Excel data sheet will display only the row of data starting with “$IIGGA”. Column defines C as hour, D as minute, E as second, G as degree of latitude, H as minute of latitude, J as degree of longitude, K as minute of longitude.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	$IIZDA,					,		,04,201	4,-	07,	00*70		
3	$IIGGA,	","	2	51	31	","	12	34.8355	"","N","	101	22.7301	,E,1,09,1.1,0.7,M,,,,*06	
8	$IIGGA,	","	2	51	32	","	12	34.8354	"","N","	101	22.7295	,E,1,09,1.1,0.3,M,,,,*0C	
13	$IIGGA,	","	2	51	33	","	12	34.8351	"","N","	101	22.7289	,E,1,09,1.1,0.7,M,,,,*01	
18	$IIGGA,	","	2	51	34	","	12	34.835	"","N","	101	22.7284	,E,1,09,1.1,0.8,M,,,,*05	
23	$IIGGA,	","	2	51	35	","	12	34.8347	"","N","	101	22.7279	,E,1,09,1.1,0.6,M,,,,*0E	
28	$IIGGA,	","	2	51	36	","	12	34.8343	"","N","	101	22.7276	,E,1,09,1.1,1.1,M,,,,*00	
33	$IIGGA,	","	2	51	37	","	12	34.834	"","N","	101	22.7273	,E,1,09,1.1,1.4,M,,,,*02	
38	$IIGGA,	","	2	51	38	","	12	34.8335	"","N","	101	22.7271	,E,1,09,1.1,1.4,M,,,,*0D	
43	$IIGGA,	","	2	51	39	","	12	34.833	"","N","	101	22.7269	,E,1,09,1.1,1.6,M,,,,*02	
48	$IIGGA,	","	2	51	40	","	12	34.8325	"","N","	101	22.7268	,E,1,09,1.1,1.6,M,,,,*09	
53	$IIGGA,	","	2	51	40	","	12	34.8325	"","N","	101	22.7268	,E,1,09,1.1,1.6,M,,,,*09	
58	$IIGGA,	","	2	51	42	","	12	34.8315	"","N","	101	22.7268	,E,1,09,1.1,1.4,M,,,,*0A	
63	$IIGGA,	","	2	51	43	","	12	34.8309	"","N","	101	22.7268	,E,1,09,1.1,1.7,M,,,,*05	
68	$IIGGA,	","	2	51	44	","	12	34.8304	"","N","	101	22.727	,E,1,09,1.1,1.3,M,,,,*02	
73	$IIGGA,	","	2	51	44	","	12	34.8304	"","N","	101	22.727	,E,1,09,1.1,1.3,M,,,,*02	
78	$IIGGA,	","	2	51	46	","	12	34.8294	"","N","	101	22.7275	,E,1,09,1.1,1.5,M,,,,*0B	
83	$IIGGA,	","	2	51	47	","	12	34.8289	"","N","	101	22.7279	,E,1,09,1.1,1.6,M,,,,*09	
88	$IIGGA,	","	2	51	47	","	12	34.8289	"","N","	101	22.7279	,E,1,09,1.1,1.6,M,,,,*09	
93	$IIGGA,	","	2	51	48	","	12	34.8285	"","N","	101	22.7283	,E,1,09,1.1,1.4,M,,,,*0D	
98	$IIGGA,	","	2	51	49	","	12	34.8281	"","N","	101	22.7287	,E,1,09,1.1,1.7,M,,,,*0F	
103	$IIGGA,	","	2	51	50	","	12	34.8278	"","N","	101	22.7292	,E,1,09,1.1,1.3,M,,,,*01	
108	$IIGGA,	","	2	51	51	","	12	34.8274	"","N","	101	22.7298	,E,1,09,1.1,1.5,M,,,,*00	
113	$IIGGA,	","	2	51	51	","	12	34.8274	"","N","	101	22.7298	,E,1,09,1.1,1.5,M,,,,*00	
118	$IIGGA,	","	2	51	51	","	12	34.8274	"","N","	101	22.7298	,E,1,09,1.1,1.5,M,,,,*00	

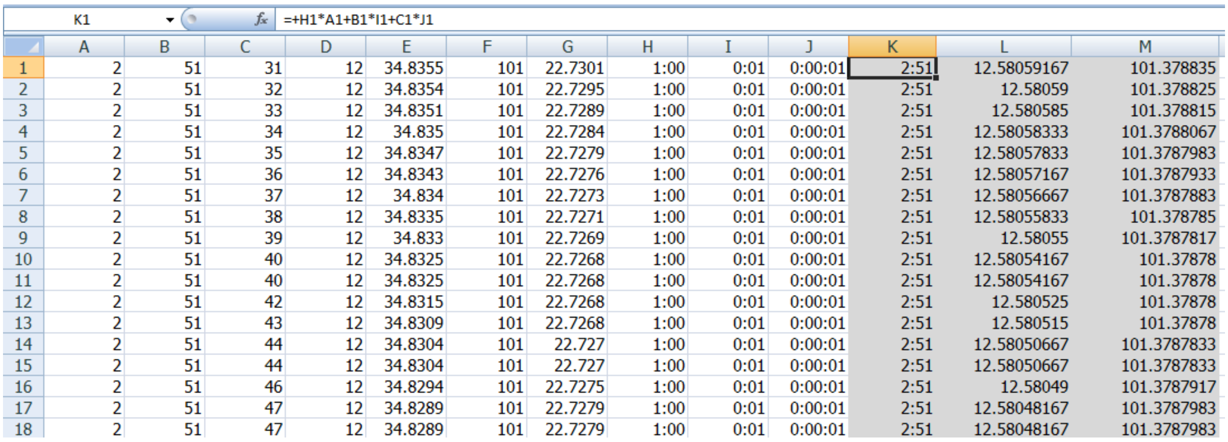
Copy all the data to a new work sheet.

g. Delete data in column A, B, F, I and L-N.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	"$IIGGA	,"	2	51		","		34.8355	","N","	1	22.7301	",E,1,09	1,0.7,M,,,,*06"	
6	"$IIGGA	,"	2	51	32	","	12	34.8354	","N","	101	22.7295	",E,1,09,1.1,0.3,M,,,,*0C"		
11	"$IIGGA	,"	2	51	33	","	12	34.8351	","N","	101	22.7289	",E,1,09,1.1,0.7,M,,,,*01"		
16	"$IIGGA	,"	2	51	34	","	12	34.835	","N","	101	22.7284	",E,1,09,1.1,0.8,M,,,,*05"		
21	"$IIGGA	,"	2	51	35	","	12	34.8347	","N","	101	22.7279	",E,1,09,1.1,0.6,M,,,,*0E"		
26	"$IIGGA	,"	2	51	36	","	12	34.8343	","N","	101	22.7276	",E,1,09,1.1,1.1,M,,,,*00"		
31	"$IIGGA	,"	2	51	37	","	12	34.834	","N","	101	22.7273	",E,1,09,1.1,1.4,M,,,,*02"		
36	"$IIGGA	,"	2	51	38	","	12	34.8335	","N","	101	22.7271	",E,1,09,1.1,1.4,M,,,,*0D"		
41	"$IIGGA	,"	2	51	39	","	12	34.833	","N","	101	22.7269	",E,1,09,1.1,1.6,M,,,,*02"		
46	"$IIGGA	,"	2	51	40	","	12	34.8325	","N","	101	22.7268	",E,1,09,1.1,1.6,M,,,,*09"		
51	"$IIGGA	,"	2	51	40	","	12	34.8325	","N","	101	22.7268	",E,1,09,1.1,1.6,M,,,,*09"		
56	"$IIGGA	,"	2	51	42	","	12	34.8315	","N","	101	22.7268	",E,1,09,1.1,1.4,M,,,,*0A"		
61	"$IIGGA	,"	2	51	43	","	12	34.8309	","N","	101	22.7268	",E,1,09,1.1,1.7,M,,,,*05"		
66	"$IIGGA	,"	2	51	44	","	12	34.8304	","N","	101	22.727	",E,1,09,1.1,1.3,M,,,,*02"		
71	"$IIGGA	,"	2	51	44	","	12	34.8304	","N","	101	22.727	",E,1,09,1.1,1.3,M,,,,*02"		
76	"$IIGGA	,"	2	51	46	","	12	34.8294	","N","	101	22.7275	",E,1,09,1.1,1.5,M,,,,*0B"		
81	"$IIGGA	,"	2	51	47	","	12	34.8289	","N","	101	22.7279	",E,1,09,1.1,1.6,M,,,,*09"		
86	"$IIGGA	,"	2	51	47	","	12	34.8289	","N","	101	22.7279	",E,1,09,1.1,1.6,M,,,,*09"		
91	"$IIGGA	,"	2	51	48	","	12	34.8285	","N","	101	22.7283	",E,1,09,1.1,1.4,M,,,,*0D"		
96	"$IIGGA	,"	2	51	49	","	12	34.8281	","N","	101	22.7287	",E,1,09,1.1,1.7,M,,,,*0F"		
101	"$IIGGA	,"	2	51	50	","	12	34.8278	","N","	101	22.7292	",E,1,09,1.1,1.3,M,,,,*01"		
106	"$IIGGA	,"	2	51	51	","	12	34.8274	","N","	101	22.7298	",E,1,09,1.1,1.5,M,,,,*00"		
111	"$IIGGA	,"	2	51	51	","	12	34.8274	","N","	101	22.7298	",E,1,09,1.1,1.5,M,,,,*00"		

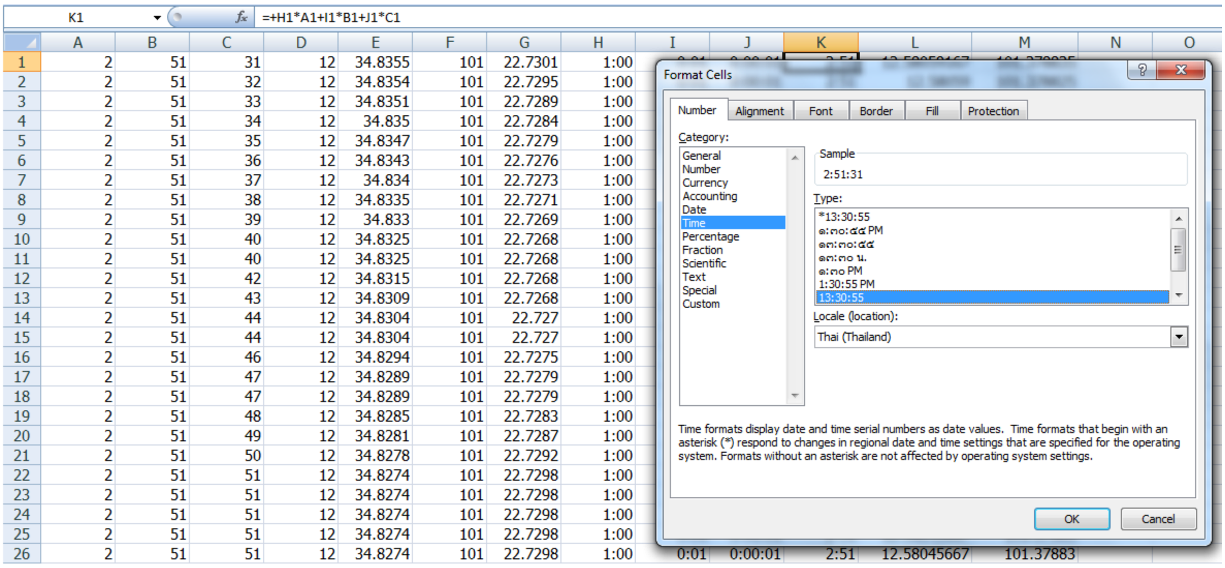
h. After deleting, calcurate UTC (Coordinated Universal Time) and degree of latitude and longitude, column H as 1:00, I as 0:01, J as 00:00:01, K as =+H1*A1+B1*I1+C1*J1, L as =+D1+E1/60 and M as =+F1+G1/60.



K1 =+H1*A1+B1*I1+C1*J1

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	2	51	31	12	34.8355	101	22.7301	1:00	0:01	0:00:01	2:51	12.58059167	101.378835
2	2	51	32	12	34.8354	101	22.7295	1:00	0:01	0:00:01	2:51	12.58059	101.378825
3	2	51	33	12	34.8351	101	22.7289	1:00	0:01	0:00:01	2:51	12.580585	101.378815
4	2	51	34	12	34.835	101	22.7284	1:00	0:01	0:00:01	2:51	12.58058333	101.3788067
5	2	51	35	12	34.8347	101	22.7279	1:00	0:01	0:00:01	2:51	12.58057833	101.3787983
6	2	51	36	12	34.8343	101	22.7276	1:00	0:01	0:00:01	2:51	12.58057167	101.3787933
7	2	51	37	12	34.834	101	22.7273	1:00	0:01	0:00:01	2:51	12.58056667	101.3787883
8	2	51	38	12	34.8335	101	22.7271	1:00	0:01	0:00:01	2:51	12.58055833	101.378785
9	2	51	39	12	34.833	101	22.7269	1:00	0:01	0:00:01	2:51	12.58055	101.3787817
10	2	51	40	12	34.8325	101	22.7268	1:00	0:01	0:00:01	2:51	12.58054167	101.37878
11	2	51	40	12	34.8325	101	22.7268	1:00	0:01	0:00:01	2:51	12.58054167	101.37878
12	2	51	42	12	34.8315	101	22.7268	1:00	0:01	0:00:01	2:51	12.580525	101.37878
13	2	51	43	12	34.8309	101	22.7268	1:00	0:01	0:00:01	2:51	12.580515	101.37878
14	2	51	44	12	34.8304	101	22.727	1:00	0:01	0:00:01	2:51	12.58050667	101.3787833
15	2	51	44	12	34.8304	101	22.727	1:00	0:01	0:00:01	2:51	12.58050667	101.3787833
16	2	51	46	12	34.8294	101	22.7275	1:00	0:01	0:00:01	2:51	12.58049	101.3787917
17	2	51	47	12	34.8289	101	22.7279	1:00	0:01	0:00:01	2:51	12.58048167	101.3787983
18	2	51	47	12	34.8289	101	22.7279	1:00	0:01	0:00:01	2:51	12.58048167	101.3787983

i. Adjust the value of UTC in columm K by selecting, “Format Cells”. “Time”, and “13:30:55”.



K1 =+H1*A1+I1*B1+J1*C1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	2	51	31	12	34.8355	101	22.7301	1:00							
2	2	51	32	12	34.8354	101	22.7295	1:00							
3	2	51	33	12	34.8351	101	22.7289	1:00							
4	2	51	34	12	34.835	101	22.7284	1:00							
5	2	51	35	12	34.8347	101	22.7279	1:00							
6	2	51	36	12	34.8343	101	22.7276	1:00							
7	2	51	37	12	34.834	101	22.7273	1:00							
8	2	51	38	12	34.8335	101	22.7271	1:00							
9	2	51	39	12	34.833	101	22.7269	1:00							
10	2	51	40	12	34.8325	101	22.7268	1:00							
11	2	51	40	12	34.8325	101	22.7268	1:00							
12	2	51	42	12	34.8315	101	22.7268	1:00							
13	2	51	43	12	34.8309	101	22.7268	1:00							
14	2	51	44	12	34.8304	101	22.727	1:00							
15	2	51	44	12	34.8304	101	22.727	1:00							
16	2	51	46	12	34.8294	101	22.7275	1:00							
17	2	51	47	12	34.8289	101	22.7279	1:00							
18	2	51	47	12	34.8289	101	22.7279	1:00							
19	2	51	48	12	34.8285	101	22.7283	1:00							
20	2	51	49	12	34.8281	101	22.7287	1:00							
21	2	51	50	12	34.8278	101	22.7292	1:00							
22	2	51	51	12	34.8274	101	22.7298	1:00							
23	2	51	51	12	34.8274	101	22.7298	1:00							
24	2	51	51	12	34.8274	101	22.7298	1:00							
25	2	51	51	12	34.8274	101	22.7298	1:00							
26	2	51	51	12	34.8274	101	22.7298	1:00	0:01	0:00:01	2:51	12.58045667	101.37883		

Format Cells

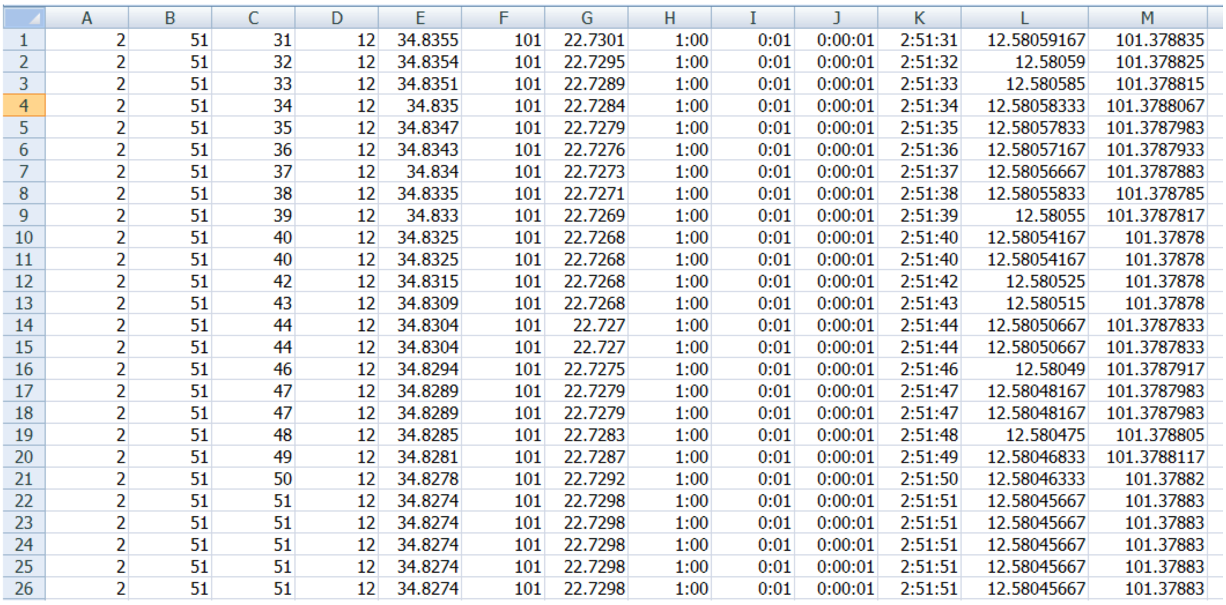
Sample: 2:51:31

Type:
*13:30:55
๑:๓๐:๕๕ PM
๑๓:๓๐:๕๕
๑๓:๓๐ น.
๑:๓๐ PM
1:30:55 PM
13:30:55

Locale (location): Thai (Thailand)

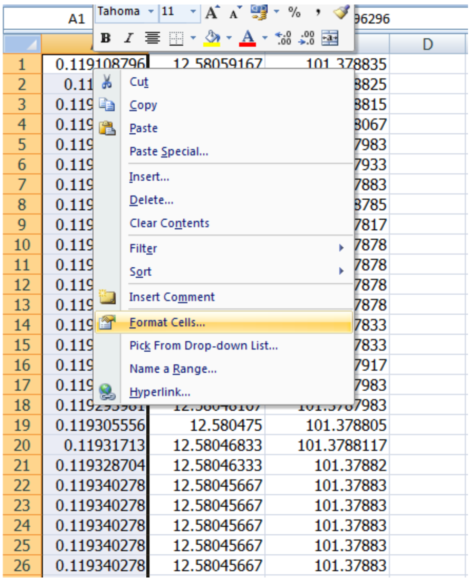
Time formats display date and time serial numbers as date values. Time formats that begin with an asterisk (*) respond to changes in regional date and time settings that are specified for the operating system. Formats without an asterisk are not affected by operating system settings.

Excel sheet UTC time will show columm K

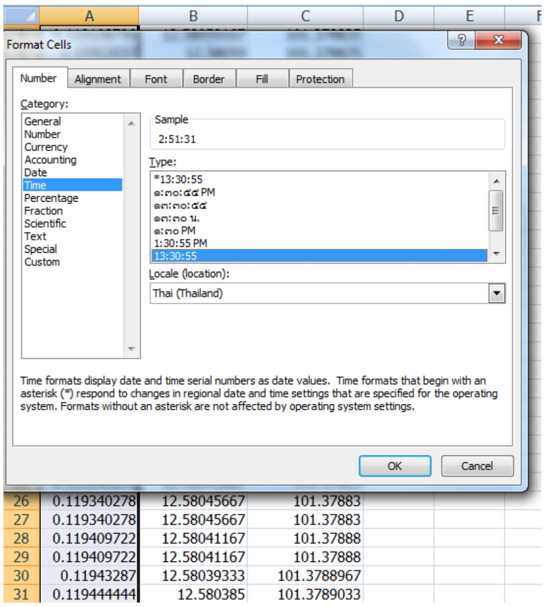


	A	B	C	D	E	F	G	H	I	J	K	L	M
1	2	51	31	12	34.8355	101	22.7301	1:00	0:01	0:00:01	2:51:31	12.58059167	101.378835
2	2	51	32	12	34.8354	101	22.7295	1:00	0:01	0:00:01	2:51:32	12.58059	101.378825
3	2	51	33	12	34.8351	101	22.7289	1:00	0:01	0:00:01	2:51:33	12.580585	101.378815
4	2	51	34	12	34.835	101	22.7284	1:00	0:01	0:00:01	2:51:34	12.58058333	101.3788067
5	2	51	35	12	34.8347	101	22.7279	1:00	0:01	0:00:01	2:51:35	12.58057833	101.3787983
6	2	51	36	12	34.8343	101	22.7276	1:00	0:01	0:00:01	2:51:36	12.58057167	101.3787933
7	2	51	37	12	34.834	101	22.7273	1:00	0:01	0:00:01	2:51:37	12.58056667	101.3787883
8	2	51	38	12	34.8335	101	22.7271	1:00	0:01	0:00:01	2:51:38	12.58055833	101.378785
9	2	51	39	12	34.833	101	22.7269	1:00	0:01	0:00:01	2:51:39	12.58055	101.3787817
10	2	51	40	12	34.8325	101	22.7268	1:00	0:01	0:00:01	2:51:40	12.58054167	101.37878
11	2	51	40	12	34.8325	101	22.7268	1:00	0:01	0:00:01	2:51:40	12.58054167	101.37878
12	2	51	42	12	34.8315	101	22.7268	1:00	0:01	0:00:01	2:51:42	12.580525	101.37878
13	2	51	43	12	34.8309	101	22.7268	1:00	0:01	0:00:01	2:51:43	12.580515	101.37878
14	2	51	44	12	34.8304	101	22.727	1:00	0:01	0:00:01	2:51:44	12.58050667	101.3787833
15	2	51	44	12	34.8304	101	22.727	1:00	0:01	0:00:01	2:51:44	12.58050667	101.3787833
16	2	51	46	12	34.8294	101	22.7275	1:00	0:01	0:00:01	2:51:46	12.58049	101.3787917
17	2	51	47	12	34.8289	101	22.7279	1:00	0:01	0:00:01	2:51:47	12.58048167	101.3787983
18	2	51	47	12	34.8289	101	22.7279	1:00	0:01	0:00:01	2:51:47	12.58048167	101.3787983
19	2	51	48	12	34.8285	101	22.7283	1:00	0:01	0:00:01	2:51:48	12.580475	101.378805
20	2	51	49	12	34.8281	101	22.7287	1:00	0:01	0:00:01	2:51:49	12.58046833	101.3788117
21	2	51	50	12	34.8278	101	22.7292	1:00	0:01	0:00:01	2:51:50	12.58046333	101.37882
22	2	51	51	12	34.8274	101	22.7298	1:00	0:01	0:00:01	2:51:51	12.58045667	101.37883
23	2	51	51	12	34.8274	101	22.7298	1:00	0:01	0:00:01	2:51:51	12.58045667	101.37883
24	2	51	51	12	34.8274	101	22.7298	1:00	0:01	0:00:01	2:51:51	12.58045667	101.37883
25	2	51	51	12	34.8274	101	22.7298	1:00	0:01	0:00:01	2:51:51	12.58045667	101.37883
26	2	51	51	12	34.8274	101	22.7298	1:00	0:01	0:00:01	2:51:51	12.58045667	101.37883

j. Copy values from column K to M to a new sheet. “Paste”, “Paste Values”. Convert the data of column A to time by “Format Cells”, “Time”, “13:30:55”.



	A	B	C
1	0.119108796	12.58059167	101.378835
19	0.119305556	12.580475	101.378805
20	0.11931713	12.58046833	101.3788117
21	0.119328704	12.58046333	101.37882
22	0.119340278	12.58045667	101.37883
23	0.119340278	12.58045667	101.37883
24	0.119340278	12.58045667	101.37883
25	0.119340278	12.58045667	101.37883
26	0.119340278	12.58045667	101.37883



Sample: 2:51:31

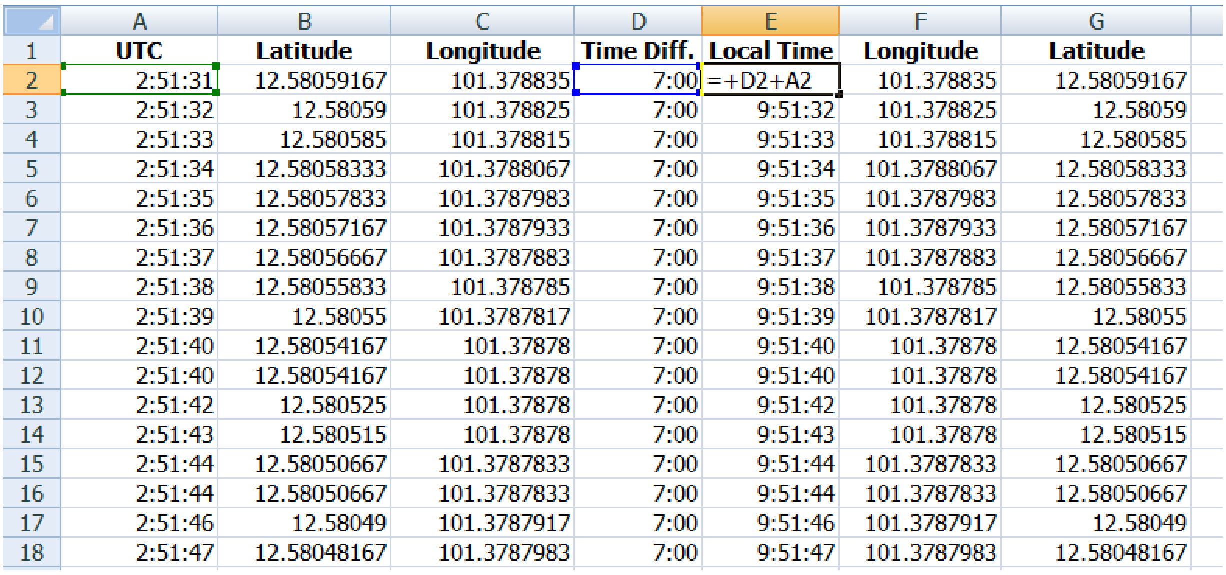
Type: 13:30:55

Locale (location): Thai (Thailand)

Time formats display date and time serial numbers as date values. Time formats that begin with an asterisk (*) respond to changes in regional date and time settings that are specified for the operating system. Formats without an asterisk are not affected by operating system settings.

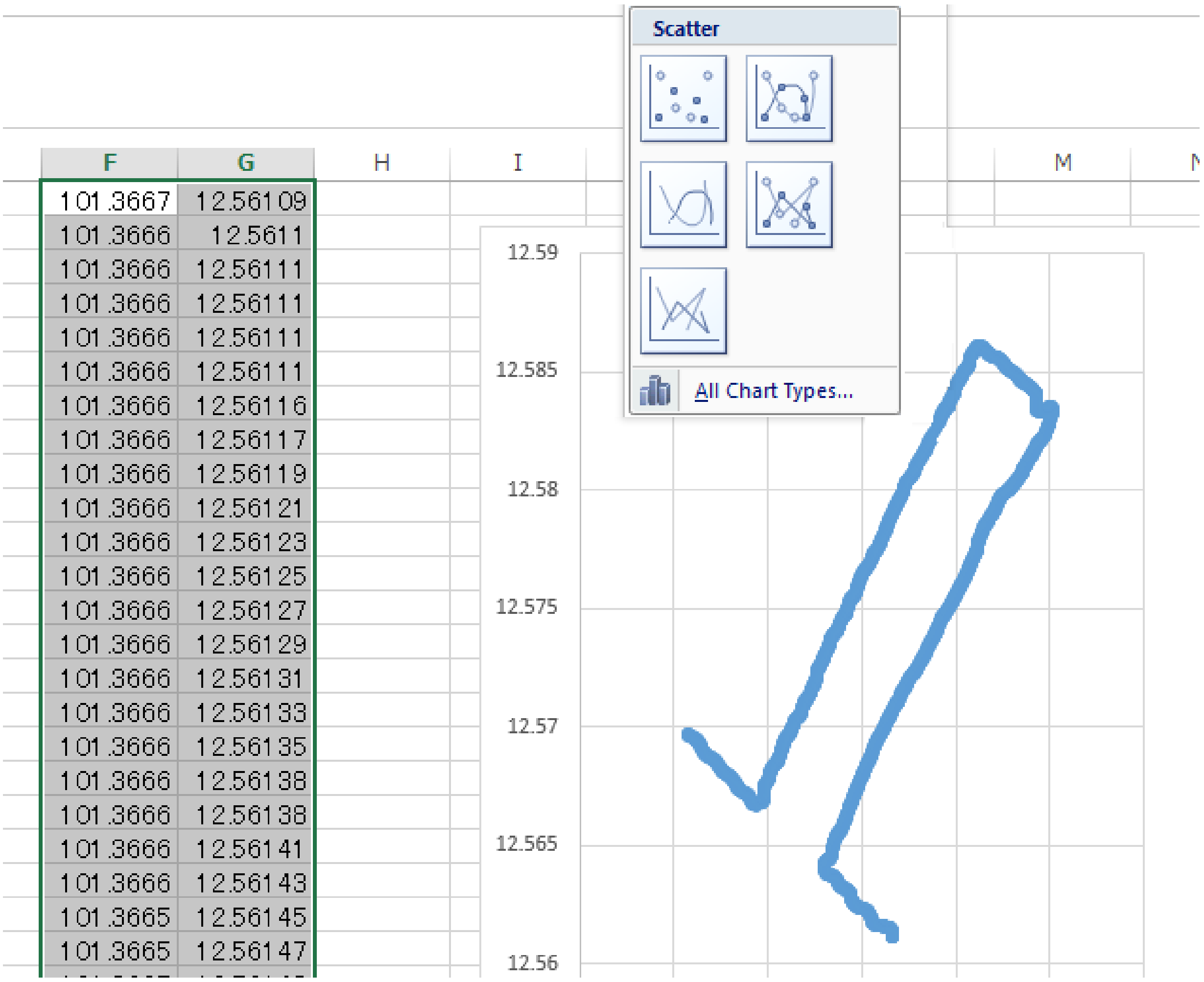
	A	B	C
26	0.119340278	12.58045667	101.37883
27	0.119340278	12.58045667	101.37883
28	0.119409722	12.58041167	101.37888
29	0.119409722	12.58041167	101.37888
30	0.11943287	12.58039333	101.3788967
31	0.119444444	12.580385	101.3789033

k. The column A is UTC. Normally, local mean time is used in the resarch. So the local time is shown in column E (UTC add Time difference). The columns of latitude and longitude should be reversed, because the latitude is on the y axis and longitude on the x axis.



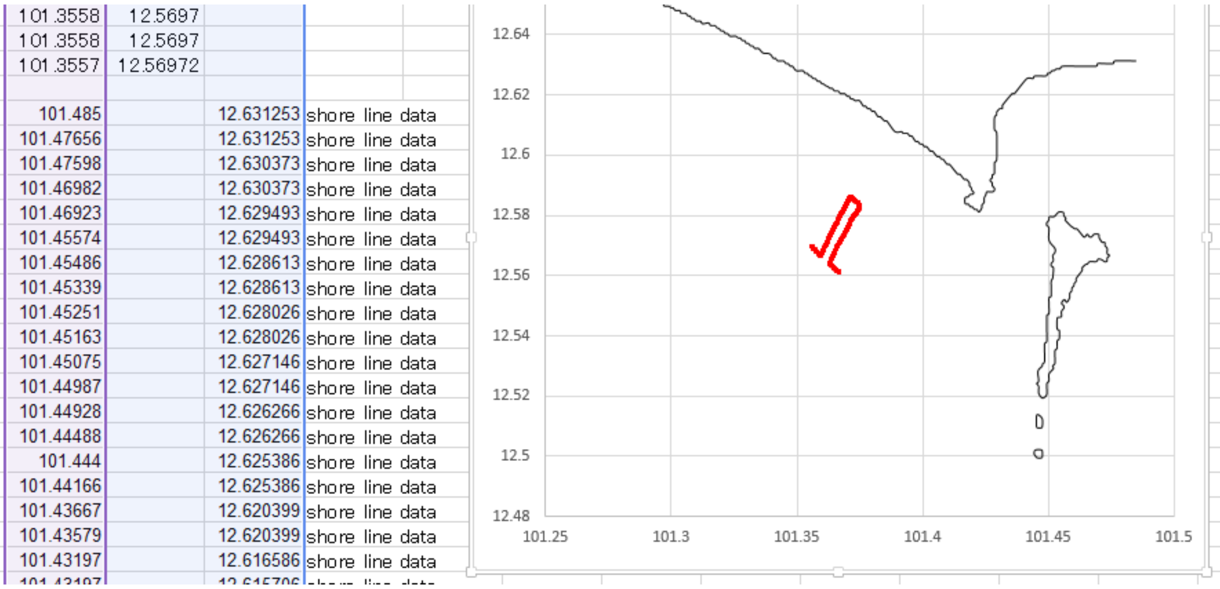
	A	B	C	D	E	F	G
1	UTC	Latitude	Longitude	Time Diff.	Local Time	Longitude	Latitude
2	2:51:31	12.58059167	101.378835	7:00	=+D2+A2	101.378835	12.58059167
3	2:51:32	12.58059	101.378825	7:00	9:51:32	101.378825	12.58059
4	2:51:33	12.580585	101.378815	7:00	9:51:33	101.378815	12.580585
5	2:51:34	12.58058333	101.3788067	7:00	9:51:34	101.3788067	12.58058333
6	2:51:35	12.58057833	101.3787983	7:00	9:51:35	101.3787983	12.58057833
7	2:51:36	12.58057167	101.3787933	7:00	9:51:36	101.3787933	12.58057167
8	2:51:37	12.58056667	101.3787883	7:00	9:51:37	101.3787883	12.58056667
9	2:51:38	12.58055833	101.378785	7:00	9:51:38	101.378785	12.58055833
10	2:51:39	12.58055	101.3787817	7:00	9:51:39	101.3787817	12.58055
11	2:51:40	12.58054167	101.37878	7:00	9:51:40	101.37878	12.58054167
12	2:51:40	12.58054167	101.37878	7:00	9:51:40	101.37878	12.58054167
13	2:51:42	12.580525	101.37878	7:00	9:51:42	101.37878	12.580525
14	2:51:43	12.580515	101.37878	7:00	9:51:43	101.37878	12.580515
15	2:51:44	12.58050667	101.3787833	7:00	9:51:44	101.3787833	12.58050667
16	2:51:44	12.58050667	101.3787833	7:00	9:51:44	101.3787833	12.58050667
17	2:51:46	12.58049	101.3787917	7:00	9:51:46	101.3787917	12.58049
18	2:51:47	12.58048167	101.3787983	7:00	9:51:47	101.3787983	12.58048167

l. Select Columns F and G, select insert chart, select a scatter plot chart. This procedure can produce a drawing of the track line.



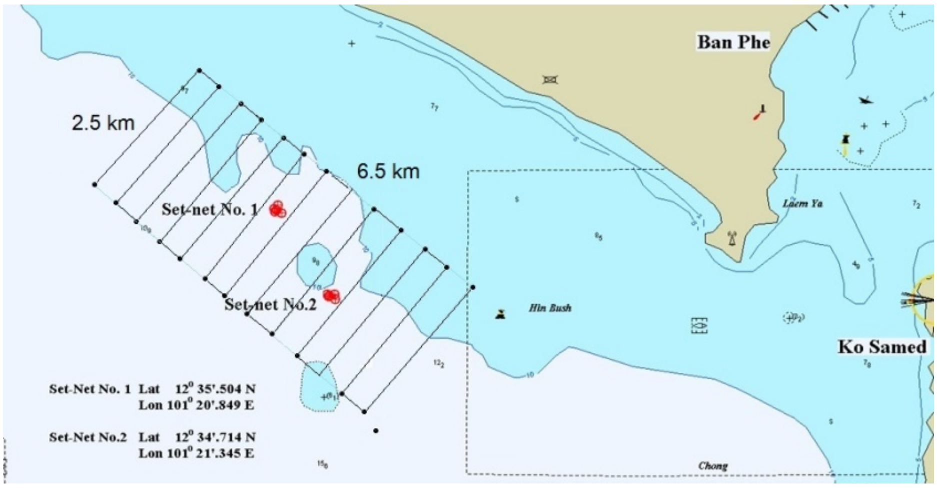
F	G
101.3667	12.56109
101.3666	12.5611
101.3666	12.56111
101.3666	12.56111
101.3666	12.56111
101.3666	12.56111
101.3666	12.56116
101.3666	12.56117
101.3666	12.56119
101.3666	12.56121
101.3666	12.56123
101.3666	12.56125
101.3666	12.56127
101.3666	12.56129
101.3666	12.56131
101.3666	12.56133
101.3666	12.56135
101.3666	12.56138
101.3666	12.56138
101.3666	12.56141
101.3666	12.56143
101.3665	12.56145
101.3665	12.56147

m. Add shore line data and detect with the tracking data. This procedure will produces a rough survey map. After a chart is added, some of the default elements should be modified to create an exquisite eye-catching map.



101.3558	12.5697		
101.3558	12.5697		
101.3557	12.56972		
101.485		12.631253	shore line data
101.47656		12.631253	shore line data
101.47598		12.630373	shore line data
101.46982		12.630373	shore line data
101.46923		12.629493	shore line data
101.45574		12.629493	shore line data
101.45486		12.628613	shore line data
101.45339		12.628613	shore line data
101.45251		12.628026	shore line data
101.45163		12.628026	shore line data
101.45075		12.627146	shore line data
101.44987		12.627146	shore line data
101.44928		12.626266	shore line data
101.44488		12.626266	shore line data
101.444		12.625386	shore line data
101.44166		12.625386	shore line data
101.43667		12.620399	shore line data
101.43579		12.620399	shore line data
101.43197		12.616586	shore line data
101.43197		12.615706	shore line data

Plotting of survey data cruise tract could be displayed to fit with the design survey transect in the study area. This experiment was conducted in the set-net fishing ground of Banphe, Rayong Province, Thailand. The coverage survey area is 2.5 x 6.5 square kilometers with parallel cruise tract of 500 meters apart.



Survey transect coverage in the set-net fishing ground of Banphe, Rayong Province, Thailand.

ANNEX I

Source Code of Program “FishFinder, Version 3.2” in Visual C

<Original source code can be downloaded from RIHN archive database:
http://www.chikyu.ac.jp/rihn/archive_datebase/archive/index.html >

```
/**************************************************************************
     * Program

**************************************************************************/

using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace jPicoFishfinder1._0
{
    struct ChannelSettings
    {
        public bool DCcoupled;
        public Imports.Range range;
        public bool enabled;
    }

    class Pwq
    {
        public Imports.PwqConditions[] conditions;
        public short nConditions;
        public Imports.ThresholdDirection direction;
        public uint lower;
        public uint upper;
        public Imports.PulseWidthType type;

        public Pwq(Imports.PwqConditions[] condit,ions
          shor   t nConditions,
            Imports.ThresholdDirectio ndirection,
            uint lower, uint upper,
            Imports.PulseWidthType ty(pe
        {
            this.conditions = conditions;
            this.nConditions = nConditions;
            this.direction = direction;
            this.lower = lower;
            this.upper = upper;
            this.type = type;
        }
    }

    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);

            short handle;
            short s = tatusImports.OpenUnit(out handle);   //1. Open the oscilloscope
            if (status != 0)
            {
                MessageBox.Show("Unable to open device", "ERROR");
            }
            else
            {
                Application.Run(new Form1(handle));
//                MessageBox.Show("Close the device?");
                Imports.CloseUnit(handle);   //12. Close the oscilloscope
            }

        }
    }
}
```

```
/**************************************************************************
       * Form1
**************************************************************************/

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Threading;
using System.IO;
using Emgu.CV;
using Emgu.CV.Structure;
using System.Net;

namespace jPicoFishfinder1._0
{

   public partial class Form1 : Form
   {
      //colorbar for GPS version
      double[] m_dLevel = { 1000, 900, 850, 800, 750, 700, 650 };



      //fishfinder
      private Color[] colorbar = new Color[8] { Color.Navy, Color.Blue, Color.Aqua,
Color.LimeGreen, Color.Yellow, Color.Orange, Color.Red, Color.DarkRed };

      Thread m_threadcvImg;
      int m_cvImgStart = 0;
      Point m_pLetfTop;
      Point m_pRightBottom;
      Bitmap m_cvBmp;
      bool m_bOdd=true;

      //1.1.1
      short[] m_sPingData;

      //AD 2 SV
      double m_dSpeed ;// Double.Parse(txtSpeed.Text);

      double m_dSR ;// Double.Parse(txtSampRate.Text);

      double m_dAbsorb;// Double.Parse(txtAbsorb.Text);

      double m_dTau;// Double.Parse(txtPulseW.Text);

      double m_dVar; // m_dVar = 10 * Math.Log10(0.5 * m_dSpeed * m_dTau * m_dPhi) -
m_dK;
```

```
        //pico below
        private readonly short _handle;
        public const int BUFFER_SIZE = 800;   //sample rage ==20k, set the range here, 1 sample
==0.0375m, 1m==26.7samples  (for 1500m/s); so (30m:800samples, 40m:1068samples)
//20150323
        public const int MAX_CHANNELS = 4;
        public const int QUAD_SCOPE = 4;
        public const int DUAL_SCOPE = 2;

        uint _timebase=1001;
        int _timeInterval;
        bool _logFlag = false;
        bool _saveFlag = false;

        short _oversample = 1;
        bool _scaleVoltages = true;

        ushort[] inputRanges = { 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000, 20000,
50000 };
        bool _ready = false;
        short _trig = 0;
        uint _trigAt = 0;
        int _sampleCount = 0;
        uint _startIndex = 0;
        bool _autoStop;
        private ChannelSettings[] _channelSettings;
        private int _channelCount;
        private Imports.Range _firstRange;
        private Imports.Range _lastRange;
        private Imports.ps4000BlockReady _callbackDelegate;
        long second = 10;

        Thread m_threadLoggin;

        public Form1(short handle)
        {
            _handle = handle;
            InitializeComponent();
            GetDeviceInfo();
            pictureBox_echograph.BackColor = Color.DarkBlue;
    //      m_cvBmp = new Bitmap(pictureBox_echograph.Width, pictureBox_echograph.Height);
        }


/************************************************************************
         * Initialise unit' structure with Variant specific defaults

************************************************************************/
        void GetDeviceInfo()
        {
            int variant = 0;
         string[] description = {
                    "Driver Version    ",
                    "USB Version       ",
                    "Hardware Version  ",
                    "Variant Info      ",
                    "Serial           ",
                    "Cal Date          ",
                    "Kernel Ver        "
                  };
         System.Text.StringBuilder line = new System.Text.StringBuilder(80);

         if (_handle >= 0)
         {
            string srLable2="";
            textBox1.Clear();
            for (int i = 0; i < 7; i++)
            {
               short requiredSize;
               Imports.GetUnitInfo(_handle, line, 80, out requiredSize, i);
               if (i == 3)
               {
                  variant = Convert.ToInt16(line.ToString());
               }
               srLable2 = String.Format("{0}: {1}", description[i], line);
               srLable2 = srLable2 + "\r\n";
               textBox1.AppendText(srLable2);
            }
            srLable2 = String.Format("Sampling Rate     : {0} ksps", 1000000/(50*(_timebase-
1)));

            textBox1.AppendText(srLable2);

            switch (variant)
            {
               case (int)Imports.Model.PS4223:
                  _firstRange = Imports.Range.Range_50MV;
                  _lastRange = Imports.Range.Range_100V;
                  _channelCount = DUAL_SCOPE;
                  break;

               case (int)Imports.Model.PS4224:
                  _firstRange = Imports.Range.Range_50MV;
                  _lastRange = Imports.Range.Range_20V;
                  _channelCount = DUAL_SCOPE;
                  break;

               case (int)Imports.Model.PS4423:
                  _firstRange = Imports.Range.Range_50MV;
                  _lastRange = Imports.Range.Range_100V;
                  _channelCount = QUAD_SCOPE;
                  break;

               case (int)Imports.Model.PS4424:
                  _firstRange = Imports.Range.Range_50MV;
```

```
                        _lastRange = Imports.Range.Range_20V;
                        _channelCount = QUAD_SCOPE;
                        break;

                    case (int)Imports.Model.PS4226:
                        _firstRange = Imports.Range.Range_50MV;
                        _lastRange = Imports.Range.Range_20V;
                        _channelCount = DUAL_SCOPE;
                        break;

                    case (int)Imports.Model.PS4227:
                        _firstRange = Imports.Range.Range_50MV;
                        _lastRange = Imports.Range.Range_20V;
                        _channelCount = DUAL_SCOPE;
                        break;

                    case (int)Imports.Model.PS4262:
                        _firstRange = Imports.Range.Range_10MV;
                        _lastRange = Imports.Range.Range_20V;
                        _channelCount = DUAL_SCOPE;
                        break;
                }
            }
        }


/****************************************************************************
         * Callback
         * used by PS4000 data block collection calls, on receipt of data.
         * used to set global flags etc checked by user routines

****************************************************************************/
        void BlockCallback(short handle, short status, IntPtr pVoid)
        {
            // flag to say done reading data
            _ready = true;
        }


/****************************************************************************
         * SetDefaults - restore default settings

****************************************************************************/
        void SetDefaults()
        {
            for (int i = 0; i < _channelCount; i++) // reset channels to most recent settings
            {
                Imports.SetChannel(_handle, Imports.Channel.ChannelA + i,
                                   (short)(_channelSettings[(int)(Imports.Channel.ChannelA + i)].enabled ?
1 : 0),
                                   (short)(_channelSettings[(int)(Imports.Channel.ChannelA +
i)].DCcoupled ? 1 : 0),
                    _channelSettings[(int)(Imports.Channel.ChannelA + i)].range);
          }
      }


/*************************************************************************
        *
        * Select _timebase, set _oversample to on and time units as nano seconds
        *

*************************************************************************/
      void SetTimebase(uint timebase)
      {
          _timebase = timebase;
          int maxSamples;

          while (Imports.GetTimebase(_handle, _timebase, BUFFER_SIZE, out _timeInterval, 1, out
maxSamples, 0) != 0)
          {
              _timebase++;
          }
          _oversample = 1;
      }


/*************************************************************************
       *  SetTrigger
       *  this function sets all the required trigger parameters, and calls the
       *  triggering functions

*************************************************************************/
      short SetTrigger(Imports.TriggerChannelProperties[] channelProperties, short
nChannelProperties, Imports.TriggerConditions[] triggerConditions, short nTriggerConditions,
Imports.ThresholdDirection[] directions, Pwq pwq, uint delay, short auxOutputEnabled, int
autoTriggerMs)
      {
          short status;

          if (
            (status =
             Imports.SetTriggerChannelProperties(_handle, channelProperties, nChannelProperties,
auxOutputEnabled,
                                          autoTriggerMs)) != 0)
          {
              return status;
          }

          if ((status = Imports.SetTriggerChannelConditions(_handle, triggerConditions,
nTriggerConditions)) != 0)
          {
              return status;
          }
```

```
            if (directions == null) directions = new Imports.ThresholdDirection[]
{ Imports.ThresholdDirection.None,
        Imports.ThresholdDirection.None, Imports.ThresholdDirection.None,
Imports.ThresholdDirection.None,
        Imports.ThresholdDirection.None, Imports.ThresholdDirection.None};

            if ((status = Imports.SetTriggerChannelDirections(_handle,
                                                   directions[(int)Imports.Channel.ChannelA],
                                                   directions[(int)Imports.Channel.ChannelB],
                                                   directions[(int)Imports.Channel.ChannelC],
                                                   directions[(int)Imports.Channel.ChannelD],
                                                   directions[(int)Imports.Channel.External],
                                                   directions[(int)Imports.Channel.Aux])) != 0)
            {
                return status;
            }

            if ((status = Imports.SetTriggerDelay(_handle, delay)) != 0)
            {
                return status;
            }

            if (pwq == null) pwq = new Pwq(null, 0, Imports.ThresholdDirection.None, 0, 0,
Imports.PulseWidthType.None);

            status = Imports.SetPulseWidthQualifier(_handle, pwq.conditions,
                                            pwq.nConditions, pwq.direction,
                                            pwq.lower, pwq.upper, pwq.type);

            return status;
        }


/***********************************************************************************
         * adc_to_mv
         *
         * Convert an 16-bit ADC count into millivolts

***********************************************************************************/
        int adc_to_mv(int raw, int ch)
        {
            return (raw * inputRanges[ch]) / Imports.MaxValue;
        }


/***********************************************************************************
         * mv_to_adc
         *
         * Convert a millivolt value into a 16-bit ADC count
         *
         *  (useful for setting trigger thresholds)
```

```
****************************************************************************/
      short mv_to_adc(short mv, short ch)
      {
         return (short)((mv * Imports.MaxValue) / inputRanges[ch]);
      }


/****************************************************************************
     * CollectBlockTriggered
     *  this function demonstrates how to collect a single block of data from the
     *  unit, when a trigger event occurs.

****************************************************************************/
      void SetBlockTrigger()
      {
         short triggerVoltage = mv_to_adc(1000,
(short)_channelSettings[(int)Imports.Channel.ChannelA].range); // ChannelInfo stores ADC counts
         Imports.TriggerChannelProperties[] sourceDetails = new
Imports.TriggerChannelProperties[] {
      new Imports.TriggerChannelProperties(triggerVoltage,
                                  256*10,
                                  triggerVoltage,
                                  256*10,
                                  Imports.Channel.ChannelA,
                                  Imports.ThresholdMode.Level)};

         Imports.TriggerConditions[] conditions = new Imports.TriggerConditions[] {
           new Imports.TriggerConditions(Imports.TriggerState.True,
                                  Imports.TriggerState.DontCare,
                                  Imports.TriggerState.DontCare,
                                  Imports.TriggerState.DontCare,
                                  Imports.TriggerState.DontCare,
                                  Imports.TriggerState.DontCare,
                                  Imports.TriggerState.DontCare)};

         Imports.ThresholdDirection[] directions = new Imports.ThresholdDirection[]
                                       { Imports.ThresholdDirection.Rising,
                                  Imports.ThresholdDirection.None,
                                  Imports.ThresholdDirection.None,
                                  Imports.ThresholdDirection.None,
                                  Imports.ThresholdDirection.None,
                                  Imports.ThresholdDirection.None };
         SetDefaults();
         /* Trigger enabled
          * Rising edge
          * Threshold = 1000mV */
         SetTrigger(sourceDetails, 1, conditions, 1, directions, null, 0, 0, 0);  //4. SetTrigger
      }
```

```
/**********************************************************************
      * DataLoggin - Logging Data to file

**********************************************************************/
      public void DataLoggin()
      {
         // setup devices

 //        _timebase = 1;

         _channelSettings = new ChannelSettings[MAX_CHANNELS];


         uint sampleCount = BUFFER_SIZE;
         PinnedArray<short>[] bufPinned = new PinnedArray<short>[1];

         int timeIndisposed;

         for (int i = 0; i < MAX_CHANNELS; i++)
         {
            _channelSettings[i].enabled = true;
            _channelSettings[i].DCcoupled = true;
            _channelSettings[i].range = Imports.Range.Range_5V;
         }

         SetDefaults(); //2. Select channel ranges and AC/DC coupling

         SetTimebase(1001); //3. Select timebase until the required nanoseconds per sample is
located

         SetBlockTrigger();  //4. Use the trigger setup functions to set up the trigger


         short[] buffers = new short[sampleCount];
         bufPinned[0] = new PinnedArray<short>(buffers);
         int status = Imports.SetDataBuffer(_handle, (Imports.Channel)1, buffers,
(int)sampleCount);//7. tell the driver where your memory buffer is

         if (!Directory.Exists("C:\\FishloggerData"))
            Directory.CreateDirectory("C:\\FishloggerData");
         TextWriter writer = new StreamWriter("C:\\FishloggerData\\" +
DateTime.Now.ToString("yyyyMMdd_HHmmss") + ".csv", false); //hh 12hour; HH 24hour

         //1.1 PingData
         m_sPingData = new short[sampleCount];

         double dLogData;
         double dShowData;

         while (_logFlag == true)
         {
            _ready = false;
            _callbackDelegate = BlockCallback;
            //5. Start the oscilloscope running
            Imports.RunBlock(_handle, 0, (int)sampleCount, _timebase, _oversample, out
timeIndisposed, 0, _callbackDelegate, IntPtr.Zero);

            while (!_ready) //6. wait until the oscilloscope is ready using the ps4000BlockReady
Callback
            {
               Thread.Sleep(100);
            }

            //***********write data to file******************
            if (_ready)   //
            {
               short overflow;
               Imports.GetValues(_handle, 0, ref sampleCount, 1,
Imports.DownSamplingMode.None, 0, out overflow);  //8. Transfer the block of data from the
ocilloscope
               writer.Write(DateTime.Now.ToString("yyyyMMdd_HH:mm:ss,"));
               for (int i = 0; i < sampleCount; i++)  //9 handle the data
               {

                  dLogData = adc_to_mv(bufPinned[0].Target[i],
(short)_channelSettings[(int)Imports.Channel.ChannelB].range); //voltage
                 //  dLogData = bufPinned[0].Target[i];      //AD

                  dShowData = dLogData;

                  writer.Write("{0},", dLogData);


                  if (dShowData > m_dLevel[0])
                     m_sPingData[i] = 7;
                  else if (dShowData > m_dLevel[1])
                     m_sPingData[i] = 6;
                  else if (dShowData > m_dLevel[2])
                     m_sPingData[i] = 5;
                  else if (dShowData > m_dLevel[3])
                     m_sPingData[i] = 4;
                  else if (dShowData > m_dLevel[4])
                     m_sPingData[i] = 3;
                  else if (dShowData > m_dLevel[5])
                     m_sPingData[i] = 2;
                  else if (dShowData > m_dLevel[6])
                     m_sPingData[i] = 1;
                  else
                     m_sPingData[i] = 0;

               }
               writer.WriteLine();

               //draw one Ping
            //************************

                //draw by Main thread using Invoker
                MethodInvoker mi = new MethodInvoker(this.drawEchograph);
                this.BeginInvoke(mi);

            //***************/

        }
        //**************************************************/
    }                   //10. repeat steps 5-9

    _logFlag = false;
    Imports.Stop(_handle); //11.Stop the oscilloscope
    writer.Close();
    foreach (PinnedArray<short> p in bufPinned)
    {
        if (p != null)
            p.Dispose();
    }
    _saveFlag = true;

}

private void btnDataLoggin_Click(object sender, EventArgs e)
{
    timer1.Enabled = false;

    second=0;
    label1.Text = new DateTime(second * 10000000).ToLongTimeString();

    _logFlag = true;
    btnDataLoggin.Enabled = false;
    btnStopLoggin.Enabled = true;
    m_threadLoggin = new Thread(new ThreadStart(DataLoggin));
    m_threadLoggin.Start();
    timer2.Enabled = true;
    label3.Text = "Data is logging... ... ...";
}

private void btnStopLoggin_Click(object sender, EventArgs e)
{
    timer2.Enabled = false; //no auto logging anymore

    m_threadLoggin.Suspend();
    _logFlag = false;
    btnStopLoggin.Enabled = false;
    btnDataLoggin.Enabled = true;
    m_threadLoggin.Resume();
    label3.Text = "Data logging Over";

}
```

```
        private void timer1_Tick(object sender, EventArgs e) //one second interval
        {
            second--;
            label1.Text = new DateTime(second*10000000).ToLongTimeString();

            if (second == 0)
            {
                timer1.Enabled = false;

                _logFlag = true;
                btnDataLoggin.Enabled = false;
                btnStopLoggin.Enabled = true;
                m_threadLoggin = new Thread(new ThreadStart(DataLoggin));
                m_threadLoggin.Start();
                timer2.Enabled = true;
                label3.Text = "Data is logging... ... ...";
            }
        }

        private void Form1_FormClosing(object sender, FormClosingEventArgs e)
        {
            if (MessageBox.Show("Sure to Stop Logging before closing the Application", "Attention",
  MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == DialogResult.OK)
            {

            }
            else e.Cancel=true;
        }

        private void timer2_Tick(object sender, EventArgs e) //one hour interval auto logging file
        {
            //*************Close current file first***********
            m_threadLoggin.Suspend();
            _logFlag = false;
            btnStopLoggin.Enabled = false;
            btnDataLoggin.Enabled = true;
            m_threadLoggin.Resume();
            label3.Text = "Data logging Over";

            //*************New file, restart thread**************
            while (_saveFlag == false);  //wait for the terminating of current m_threadLogging
            _saVeFlag = false; //clear;
            _logFlag = true;
            btnDataLoggin.Enabled = false;
            btnStopLoggin.Enabled = true;
            m_threadLoggin = new Thread(new ThreadStart(DataLoggin));
            m_threadLoggin.Start();
            label3.Text = "Data is logging... ... ...";
        }
        //jPicoFishfinder 1.1 add functions as below
        private void drawEchograph()
        {
            uint isamples = BUFFER_SIZE;

            Bitmap bitmap = new Bitmap(this.pictureBox_echograph.Width,
this.pictureBox_echograph.Height);
            Graphics gEcho = Graphics.FromImage(bitmap);   //this is like doublebuffer, draw on the
bitmap, and show on the picTureBox


            float y = pictureBox_echograph.Height / (float)isamples;
            float x = this.pictureBox_echograph.Width - 1;// gEcho.PageScale;
            //            int x =1311; //(840*3.125/2)

            if (this.pictureBox_echograph.Image == null)
            {
                for (int i = 0; i < isamples; i++)
                    gEcho.DrawLine(new Pen(colorbar[m_sPingData[i]]), x, i * y, x, (i + 1) * y);
            }
            else
            {
                gEcho.DrawImage(this.pictureBox_echograph.Image, -1, 0); //draw image from
current picturebox to a new position DrawImage(image,x,y),==translate
                //            gEcho.DrawImage(this.pictureBox_echograph.Image, new Rectangle(0, 0,
pictureBox_echograph.Width-1, pictureBox_echograph.Height), new Rectangle(1, 0,
this.pictureBox_echograph.Image.Width-1, this.pictureBox_echograph.Image.Height),
GraphicsUnit.Pixel);
                for (int i = 0; i < isamples; i++)
                    gEcho.DrawLine(new Pen(colorbar[m_sPingData[i]]), x, i * y, x, (i + 1) * y);
            }
            //          DrawDepthLable(gEcho, irange);  // draw depth label
            this.pictureBox_echograph.Image = bitmap; //update to the picBox,

            gEcho.Dispose();

        }

    }
}
```

```
/***************************************************************************
        * Form1_Designer
***************************************************************************/

namespace jPicoFishfinder1._0
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Form1));
            this.btnDataLoggin = new System.Windows.Forms.Button();
            this.btnStopLoggin = new System.Windows.Forms.Button();
            this.label1 = new System.Windows.Forms.Label();
            this.timer1 = new System.Windows.Forms.Timer(this.components);
            this.label2 = new System.Windows.Forms.Label();
            this.textBox1 = new System.Windows.Forms.TextBox();
            this.timer2 = new System.Windows.Forms.Timer(this.components);
            this.label3 = new System.Windows.Forms.Label();
            this.pictureBox_echograph = new System.Windows.Forms.PictureBox();
            this.pictureBox_Colorbar = new System.Windows.Forms.PictureBox();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox_echograph)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox_Colorbar)).BeginInit();
            this.SuspendLayout();
            //
            // btnDataLoggin
            //
            this.btnDataLoggin.Location = new System.Drawing.Point(940, 247);
            this.btnDataLoggin.Name = "btnDataLoggin";
            this.btnDataLoggin.Size = new System.Drawing.Size(101, 37);
            this.btnDataLoggin.TabIndex = 0;
            this.btnDataLoggin.Text = "Start Logging";
            this.btnDataLoggin.UseVisualStyleBackColor = true;
            this.btnDataLoggin.Click += new System.EventHandler(this.btnDataLoggin_Click);
            //
            // btnStopLoggin
            //
            this.btnStopLoggin.Enabled = false;
            this.btnStopLoggin.Location = new System.Drawing.Point(1055, 247);
            this.btnStopLoggin.Name = "btnStopLoggin";
            this.btnStopLoggin.Size = new System.Drawing.Size(98, 37);
            this.btnStopLoggin.TabIndex = 1;
            this.btnStopLoggin.Text = "Stop Logging";
            this.btnStopLoggin.UseVisualStyleBackColor = true;
            this.btnStopLoggin.Click += new System.EventHandler(this.btnStopLoggin_Click);
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Font = new System.Drawing.Font("SimSun", 30F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(134)));
            this.label1.Location = new System.Drawing.Point(954, 189);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(185, 40);
            this.label1.TabIndex = 2;
            this.label1.Text = "00:00:00";
            //
            // timer1
            //
            this.timer1.Enabled = true;
            this.timer1.Interval = 1000;
            this.timer1.Tick += new System.EventHandler(this.timer1_Tick);
            //
            // label2
            //
            this.label2.AutoSize = true;
            this.label2.Location = new System.Drawing.Point(934, 9);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(103, 12);
            this.label2.TabIndex = 3;
            this.label2.Text = "Device Information:\ r\ n";
            //
            // textBox1
            //
            this.textBox1.Location = new System.Drawing.Point(936, 33);
            this.textBox1.Multiline = true;
            this.textBox1.Name = "textBox1";
            this.textBox1.ReadOnly = true;
```

```
            this.textBox1.Size = new System.Drawing.Size(226, 106);
            this.textBox1.TabIndex = 4;
            this.textBox1.Text = "Driver Version    : \r\nUSB Version        : \r\nHardware Version  : 
\r\nVariant Info   " +
                "   : \r\nSerial            : \r\nCal Date          : \r\nKernel Ver        :\r\nSampling" +
                " Rate     : ";
            // 
            // timer2
            // 
            this.timer2.Interval = 3600000;
            this.timer2.Tick += new System.EventHandler(this.timer2_Tick);
            // 
            // label3
            // 
            this.label3.AutoSize = true;
            this.label3.Font = new System.Drawing.Font("SimSun", 12F, 
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));
            this.label3.Location = new System.Drawing.Point(937, 157);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(216, 16);
            this.label3.TabIndex = 5;
            this.label3.Text = "Data logging will start...";
            // 
            // pictureBox_echograph
            // 
            this.pictureBox_echograph.Location = new System.Drawing.Point(13, 13);
            this.pictureBox_echograph.Name = "pictureBox_echograph";
            this.pictureBox_echograph.Size = new System.Drawing.Size(904, 676);
            this.pictureBox_echograph.TabIndex = 6;
            this.pictureBox_echograph.TabStop = false;
            // 
            // pictureBox_Colorbar
            // 
            this.pictureBox_Colorbar.Anchor = 
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top | 
System.Windows.Forms.AnchorStyles.Bottom) 
                        | System.Windows.Forms.AnchorStyles.Right)));
            this.pictureBox_Colorbar.Image = 
((System.Drawing.Image)(resources.GetObject("pictureBox_Colorbar.Image")));
            this.pictureBox_Colorbar.InitialImage = 
((System.Drawing.Image)(resources.GetObject("pictureBox_Colorbar.InitialImage")));
            this.pictureBox_Colorbar.Location = new System.Drawing.Point(13, 172);
            this.pictureBox_Colorbar.Name = "pictureBox_Colorbar";
            this.pictureBox_Colorbar.Size = new System.Drawing.Size(24, 304);
            this.pictureBox_Colorbar.SizeMode = 
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
            this.pictureBox_Colorbar.TabIndex = 21;
            this.pictureBox_Colorbar.TabStop = false;
            // 
            // Form1
            // 
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 12F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(1165, 701);
            this.Controls.Add(this.pictureBox_Colorbar);
            this.Controls.Add(this.pictureBox_echograph);
            this.Controls.Add(this.label3);
            this.Controls.Add(this.textBox1);
            this.Controls.Add(this.label2);
            this.Controls.Add(this.label1);
            this.Controls.Add(this.btnStopLoggin);
            this.Controls.Add(this.btnDataLoggin);
            this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
            this.Name = "Form1";
            this.Text = "jPicoFishfinder_GPS1.0";
            this.FormClosing += new
System.Windows.Forms.FormClosingEventHandler(this.Form1_FormClosing);
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox_echograph)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox_Colorbar)).EndInit();
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Button btnDataLoggin;
        private System.Windows.Forms.Button btnStopLoggin;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Timer timer1;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.TextBox textBox1;
        private System.Windows.Forms.Timer timer2;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.PictureBox pictureBox_echograph;
        private System.Windows.Forms.PictureBox pictureBox_Colorbar;
    }
}
```

```
/************************************************************
*
* Filename:    PS4000Imports.cs
*
* Copyright:   Pico Technology Limited 2009
*
* Author:      MJL
*
* Description:
*   This file contains all the .NET wrapper calls needed to support
*   the console example. It also has the enums and structs required
*   by the (wrapped) function calls.
*
* History:
*     14Dec06 MJL     Created
*         15Oct09      RPM Modified for PS4000
*
* Revision Info: "file %n date %f revision %v"
*                                                    ””
************************************************************/

using System;
using System.Runtime.InteropServices;
using System.Text;

namespace jPicoFishfinder1._0
{
        class Imports
        {
                #region constants
                private const string _DRIVER_FILENAME = "ps4000.dll";

                public const int MaxValue = 32764;
                #endregion

                #region Driver enums

                public enum Channel : int
                {
                        ChannelA,
                        ChannelB,
                        ChannelC,
                        ChannelD,
                        External,
                        Aux,
                        None,
                }

                public enum Range : int
                {
                        Range_10MV,
                        Range_20MV,
                        Range_50MV,
                        Range_100MV,
                        Range_200MV,
                        Range_500MV,
                        Range_1V,
                        Range_2V,
                        Range_5V,
                        Range_10V,
                        Range_20V,
                        Range_50V,
        Range_100V,
                }

                public enum ReportedTimeUnits : int
                {
                        FemtoSeconds,
                        PicoSeconds,
                        NanoSeconds,
                        MicroSeconds,
                        MilliSeconds,
                        Seconds,
                }

                public enum ThresholdMode : int
                {
                        Level,
                        Window
                }

                public enum ThresholdDirection : int
                {
                        // Values for level threshold mode
                        //
                        Above,
                        Below,
                        Rising,
                        Falling,
                        RisingOrFalling,

                        // Values for window threshold mode
                        //
                        Inside = Above,
                        Outside = Below,
                        Enter = Rising,
                        Exit = Falling,
                        EnterOrExit = RisingOrFalling,

                        None = Rising,
                }

                public enum DownSamplingMode : int
                {
                        None,
                        Aggregate
                }

                public enum PulseWidthType : int
                {
                        None,
                        LessThan,
                        GreaterThan,
                        InRange,
                        OutOfRange
                }

                public enum TriggerState : int
                {
                        DontCare,
                        True,
                        False,
                }

        public enum Model : int
        {
        NONE = 0,
        PS4223 = 4223,
        PS4224 = 4224,
        PS4423 = 4423,
        PS4424 = 4424,
        PS4226 = 4226,
        PS4227 = 4227,
        PS4262 = 4262,
        }

#endregion

                [StructLayout(LayoutKind.Sequential, Pack = 1)]
                public struct TriggerChannelProperties
                {
                        public short ThresholdMajor;
                        public ushort HysteresisMajor;
                        public short ThresholdMinor;
                        public ushort HysteresisMinor;
                        public Channel Channel;
                        public ThresholdMode ThresholdMode;


                        public TriggerChannelProperties(
                                short thresholdMajor,
                                ushort hysteresisMajor,
                                short thresholdMinor,
                                ushort hysteresisMinor,
                                Channel channel,
                                ThresholdMode thresholdMode)
                        {
                                this.ThresholdMajor = thresholdMajor;
                                this.HysteresisMajor = hysteresisMajor;
                                this.ThresholdMinor = thresholdMinor;
                                this.HysteresisMinor = hysteresisMinor;
                                this.Channel = channel;
                                this.ThresholdMode = thresholdMode;
                        }
                }

                [StructLayout(LayoutKind.Sequential, Pack = 1)]
                public struct TriggerConditions
                {
                        public TriggerState ChannelA;
                        public TriggerState ChannelB;
                        public TriggerState ChannelC;
                        public TriggerState ChannelD;
                        public TriggerState External;
                        public TriggerState Aux;
                        public TriggerState Pwq;

                        public TriggerConditions(
                                TriggerState channelA,
                                TriggerState channelB,
                                TriggerState channelC,
                                TriggerState channelD,
                                TriggerState external,
                                TriggerState aux,
                                TriggerState pwq)
                        {
                                this.ChannelA = channelA;
```

```
			this.ChannelB = channelB;
			this.ChannelC = channelC;
			this.ChannelD = channelD;
			this.External = external;
			this.Aux = aux;
			this.Pwq = pwq;
		}
	}

	[StructLayout(LayoutKind.Sequential, Pack = 1)]
	public struct PwqConditions
	{
		public TriggerState ChannelA;
		public TriggerState ChannelB;
		public TriggerState ChannelC;
		public TriggerState ChannelD;
		public TriggerState External;
		public TriggerState Aux;

		public PwqConditions(
			TriggerState channelA,
			TriggerState channelB,
			TriggerState channelC,
			TriggerState channelD,
			TriggerState external,
			TriggerState aux)
		{
			this.ChannelA = channelA;
			this.ChannelB = channelB;
			this.ChannelC = channelC;
			this.ChannelD = channelD;
			this.External = external;
			this.Aux = aux;
		}
	}

	#region Driver Imports
	#region Callback delegates
	public delegate void ps4000BlockReady(short handle, short status, IntPtr pVoid);


	public delegate void ps4000StreamingReady(
		short handle,
		int noOfSamples,
		uint startIndex,
		short ov,
		uint triggerAt,
		short triggered,
		short autoStop,
		IntPtr pVoid);


	public delegate void ps4000DataReady(
		short handle,
            int noOfSamples,
            short overflow,
            uint triggerAt,
            short triggered,
            IntPtr pVoid);
        #endregion

        [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000OpenUnit")]
        public static extern short OpenUnit(out short handle);

        [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000CloseUnit")]
        public static extern short CloseUnit(short handle);

        [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000RunBlock")]
        public static extern short RunBlock(
            short handle,
            int noOfPreTriggerSamples,
            int noOfPostTriggerSamples,
            uint timebase,
            short oversample,
            out int timeIndisposedMs,
            ushort segmentIndex,


    ps4000BlockReady lpps4000BlockReady,
            IntPtr pVoid);

        [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000Stop")]
        public static extern short Stop(short handle);

        [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000SetChannel")]
        public static extern short SetChannel(
            short handle,
            Channel channel,
            short enabled,
            short dc,
            Range range);

        [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000SetDataBuffers")]
        public static extern short SetDataBuffers(
            short handle,
            Channel channel,
            short[] bufferMax,
            short[] bufferMin,
            int bufferLth);

    [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000SetDataBuffer")]
        public static extern short SetDataBuffer(
            short handle,
            Channel channel,
            short[] buffer,
            int bufferLth);

        [DllImport(_DRIVER_FILENAME, EntryPoint =
"ps4000SetTriggerChannelDirections")]
        public static extern short SetTriggerChannelDirections(
            short handle,

    ThresholdDirection channelA,

    ThresholdDirection channelB,

    ThresholdDirection channelC,

    ThresholdDirection channelD,

    ThresholdDirection ext,

    ThresholdDirection aux);

        [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000GetTimebase")]
        public static extern short GetTimebase(
             short handle,
             uint timebase,
             int noSamples,
             out int timeIntervalNanoseconds,
             short oversample,
             out int maxSamples,
             ushort segmentIndex);

        [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000GetValues")]
        public static extern short GetValues(
            short handle,
            uint startIndex,
            ref uint noOfSamples,
            uint downSampleRatio,
            DownSamplingMode downSampleDownSamplingMode,
            ushort segmentIndex,
            out short overflow);

        [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000SetPulseWidthQualifier")]
        public static extern short SetPulseWidthQualifier(
            short handle,
            PwqConditions[] conditions,
            short numConditions,
            ThresholdDirection direction,
            uint lower,
            uint upper,
                        PulseWidthType type);

                [DllImport(_DRIVER_FILENAME, EntryPoint =
"ps4000SetTriggerChannelProperties")]
                public static extern short SetTriggerChannelProperties(
                        short handle,
                        TriggerChannelProperties[] channelProperties,
                        short numChannelProperties,
                        short auxOutputEnable,
                        int autoTriggerMilliseconds);

                [DllImport(_DRIVER_FILENAME, EntryPoint =
"ps4000SetTriggerChannelConditions")]
                public static extern short SetTriggerChannelConditions(
                        short handle,
                        TriggerConditions[] conditions,
                        short numConditions);

                [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000SetTriggerDelay")]
                public static extern short SetTriggerDelay(short handle, uint delay);

                [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000GetUnitInfo")]
                public static extern short GetUnitInfo(short handle, StringBuilder infoString, short
stringLength, out short requiredSize, int info);

                [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000RunStreaming")]
                public static extern short RunStreaming(
                        short handle,
                        ref uint sampleInterval,
                        ReportedTimeUnits sampleIntervalTimeUnits,
                        uint maxPreTriggerSamples,
                        uint maxPostPreTriggerSamples,
                        bool autoStop,
                        uint downSamplingRation,
                        uint overviewBufferSize);

                [DllImport(_DRIVER_FILENAME, EntryPoint =
"ps4000GetStreamingLatestValues")]
                public static extern short GetStreamingLatestValues(
                        short handle,
                        ps4000StreamingReady lpps4000StreamingReady,
                        IntPtr pVoid);

                [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000SetNoOfCaptures")]
                public static extern short SetNoOfRapidCaptures(
                        short handle,
                        ushort nWaveforms);
```

```
        [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000MemorySegments")]
        public static extern short MemorySegments(
            short handle,
            ushort nSegments,
            out int nMaxSamples);

        [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000SetDataBufferBulk")]
        public static extern short SetDataBuffersRapid(
            short handle,
            Channel channel,
            short[] buffer,
            int bufferLth,
            ushort waveform);

        [DllImport(_DRIVER_FILENAME, EntryPoint = "ps4000GetValuesBulk")]
        public static extern short GetValuesRapid(
            short handle,
            ref uint noOfSamples,
            ushort fromSegmentIndex,
            ushort toSegmentIndex,
            short[] overflows);
        #endregion
    }
}
```

```
/**************************************************************************
        * PS4000Pinned Array

**************************************************************************/

using System;
using System.Runtime.InteropServices;


namespace jPicoFishfinder1
{
  public class PinnedArray<T>
  {
          GCHandle _pinnedHandle;
           private bool _disposed;

           public PinnedArray(int arraySize) : this(new T[arraySize]) { }

           public PinnedArray(T[] array)
          {
                   _pinnedHandle = GCHandle.Alloc(array, GCHandleType.Pinned);
           }

          ~PinnedArray()
           {
                   Dispose();
           }

           public T[] Target
          {
                     get { return (T[])_pinnedHandle.Target; }
           }

            public static implicit operator T[](PinnedArray<T> a)
           {
                    if (a == null)
                    return null;
                    else
                    return (T[])a._pinnedHandle.Target;
          }

           public void Dispose()
           {
                   if (!_disposed)
                    {
                             _pinnedHandle.Free();
                             _disposed = true;

                             GC.SuppressFinalize(this);
                    }
           }
  }
}
```

Acknowledgements

Our activities are supported by “Coastal Area-capability Enhancement in Southeast Asia” Project, Research Institute for Humanity and Nature (RIHN, Project No.14200061).

We would like to thank Mr. Yuzo Abe of FUSION Inc. for their technical support and Dr. Kazuo Amakasu of Tokyo University of Marine Science and Technology for his theoretical advice. Thanks to Mr. Soichiro Matsushita (currently belonging to Honda Electronics Co., Ltd.) who vigorously conducted the experiment.

Research Institute for Humanity and Nature
February 2017

ISBN 978-4-906888-35-1