PNC TN8410 89-062

JASPAS 90-1

LIMITED DISTRIBUTION

JASPAS

JAPAN SUPPORT PROGRAMME FOR AGENCY SAFEGUARDS

TASK No. : JC-7

TITLE : Automated Gravimetric Sample Preparation System (AGSS) for Safeguards Analysis

Sample Preparation System for Plutonium Product
(Interim Report of JASPAS JC-7)

January 1990

Produced by

Power Reactor & Nuclear Fuel

Development Corporation

〒319-1184　茨城県那珂郡東海村大字村松4番地49
核燃料サイクル開発機構
技術展開部　技術協力課

JASPAS90-1
LIMITED DISTRIBUTION
PNC TN8410 89-062
1989年

JASPAS JC-7; Automated gravimetric sample preparation system (AGSS) for safeguards analysis

Sample preparation system for plutonium product

(Interim report of JASPAS JC-7)

佐藤宗一　西田恭輔　阿部勝男
久野祐輔　秋山孝夫

要　　旨

〔目　　的〕

再処理工場プルトニウム製品計量槽における保障措置収去試料の採取精度の向上，及び処理の自動化を図る。

〔概　　要〕

再処理工場の核物質の計量管理は，核物質の保障措置上重要である。再処理工場では，核物質の入量及び出量計量に係るベッセルの容量測定，試料のサンプリング，希釈調製及び分析により核物質量の計量を実施してきた。計量管理における誤差は，これら各操作に存在する誤差の総和であるが，これらの誤差をさらに小さくするよう，近年IAEAから要求されていた。そのため，対IAEA保障措置技術開発支援計画（JASPAS）の一環として，特に試料調製以降の操作を対象に本題を計画し，実施した。なお，今回は出量計量（プルトニウム製品濃縮受槽）の自動重量サンプリング装置の開発を実施した。

再処理工場の出量計量に係るサンプリング，希釈調製操作は，これまでピペットやメスフラスコ等を使用する容量法で実施してきた。この方法による誤差は，グローブボックス内の特殊環境で操作するため，温度補正等を行っても，なお全体誤差に占める割合が大きいものとなっていた。そのため，温度補正等環境条件に存在しない重量法に取組み，遠隔操作性を重点に自動化装置を開発した。

主な成果は次のとおりである。

(1) 遠隔操作用ロボットアームの小型化を図り，既在のグローブボックス内への設置を可能とした。

(2) 自動化の動作シーケンスのケーススタディを行ない，最適なシーケンスを確立し，全自動の重量サンプリング装置として完成できた。

(3) 本重量サンプリング装置によるサンプリング，希釈調製，2次サンプリングの総合誤差は0.1%以下であり，良好な性能を示すことを確認した。

(4) これまで多大な労力を要していた保障措置試料の調製作業について大巾に省力化を図ることができた。

再処理工場工務部分析課

AUTOMATED GRAVIMETRIC SAMPLE PREPARATION SYSTEM (AGSS) FOR SAFEGUARDS ANALYSIS

— SAMPLE PREPARATION SYSTEM FOR PLUTONIUM PRODUCT —

(INTERIM RPORT OF JASPAS JC-7)

CONTENTS

AUTOMATED GRAVIMETRIC SAMPLE PREPARATION SYSTEM (AGSS) FOR SAFEGUARDS ANALYSIS

— SAMPLE PREPARATION SYSTEM FOR PLUTONIUM PRODUCT —

(INTERIM REPORT OF JASPAS JC-7)

S. SATO, K. NISHIDA, K. ABE, Y. KUNO, T. AKIYAMA

ABSTRACT

This report describes the results of the performance test of the automated gravimetric sample preparation system for plutonium product safeguards.

Recent accountability and safeguards analysis require more accurate sample preparation and measurement. With respect to the accountability measurement, the advanced techniques such as the controlled potential coulometry have been developed so far. As for sample preparation, several sampling devices on the volumetric sample preparation have been also developed. It has been, however, still required that sample preparation be performed on the gravimetric base in order to achieve more accurate accountability. It is expected that the errors due to the difference between the temperature on vessel volume measurement and that on quantitative analysis of sample can be eliminated with the gravimetric method.

An automated gravimetric sample preparation system for the purpose of plutonium safeguards sample preparation has been developed at the Tokai reprocessing plant. This system is capable of taking aliquots and diluting samples automatically with a robot hand. The summary results of the performance test are as follows,

i) the error estimated based on the preparation scheme is around 0.02%, which is due to the error based on the solution evaporation observed throughout the sample preparation.

ii) the time required for the safeguards sample preparation consisting of national and international inspections is within 70 minutes.

It is concluded that the system can be satisfactorily used for the routine safeguards sample preparation. This development has been made under Japan Support Program for Agency Safeguards (JASPAS)[1].

1. INTRODUCTION

As one aspect of safeguards procedures at the Tokai Reprecessing Plant (TRP), samples from the input accountability vessel and the plutonium product accountability vessel are taken and treated by the Japan government and by the International Atomic Energy Agency (actually samples are divided up and taken back to the respective laboratories). In addition, samples from these vessels are analysed for accountability purpose. The book inventory of nuclear material in the reprocessing plant is calculated from measurement data of these accountability vessels. This figure is compared with the physical inventory of nuclear material determined at the time of physical inventory taking (PIT) and the difference berween the two values is calculated to be the material unaccounted for (MUF).

The inspectorate authorities, based on this MUF value, determined whether or not a nuclear material has been diverted from the reprocessing plant. In actual practice, however, several possible sources of error can occur during sampling and treatment, and thus the MUF is not necessarily a valid figure. The inspectorates must take account of these error sources statistically in their determinations. From the standpoint of preventing illegitimate use of nuclear material, it is therefore necessary to minimize these errors as much as possible.

In general, these errors can be divided into two categories; sampling errors and analysis errors. Improvements on techniques and equipments to be adopted have decreased the portion of analysis errors in the total errors in recent years. Sampling errors, on the other hand, because of continued use of the conventioanl volume sampling method, now account for a comparatively large percentage of total errors.

Furthrmore, in order to decrease the error on sampling, switching from the conventional volumetric sampling method to a gravimetric sampling method and designing a fully automated sampling device were conducted, taking this opportunity.

Volumetric sampling method, conventionally used, is based on sample taking with a pipet, then dilution in a measuring flask. The following sources of error must be taken into consideration when this type of sampling is implemented.

- an error in determining the volume of the pipet or measuring flask.
- a change in the sample volume and vessel volume due to temperature
- an error resulting from variation of discharge quantity due to the variety of viscosity of sample
- an error resulting from variation in aligning the marked line on the pipet and flask sampling and dilution

In gravimetric sampling, on the other hand, the sample is weighed on the scale at the same time it is taken and diluted, and thus the error sources listed above are avoided, In addition, there is no need to align the marked lines, and the operation itself is thus simplified.

When gravimetric sampling data are changed to volumetric unit, it is necessary to know the density of solution, which involves a separate measurement. Even this case, it is expected, when the volumetric sampling method is replaced with the gravimetric sampling method, to reduce errors to approximately one-third.

An experimental device was constructed on October in 1984, and cold tests were carried out with this device. As a results of these cold tests it was recognised that following points should require improvement;

- a minute unsettlement in the scale due to air movement and electro-static effect

- over 20-30 minute interval between the primary sampling and secondary one, a problem arose with errors caused by evaporation of the sample solution.
- in order to facilitate maintenance when there was a breakdown, the whole system needed to be divided up into several small units.

Based upon the above results, a demonstration equipment was constructed, and capability of the equipment was tested from September 1985 until September 1986. Software improvements which should function to reduce the total time required for the operation has also been done. This report deals with the results of the performance test on this demonstration equipment.

2. COMPARISON BEWTEEN VOLUMETRIC METHOD AND GRAVIMETRIC METHOD

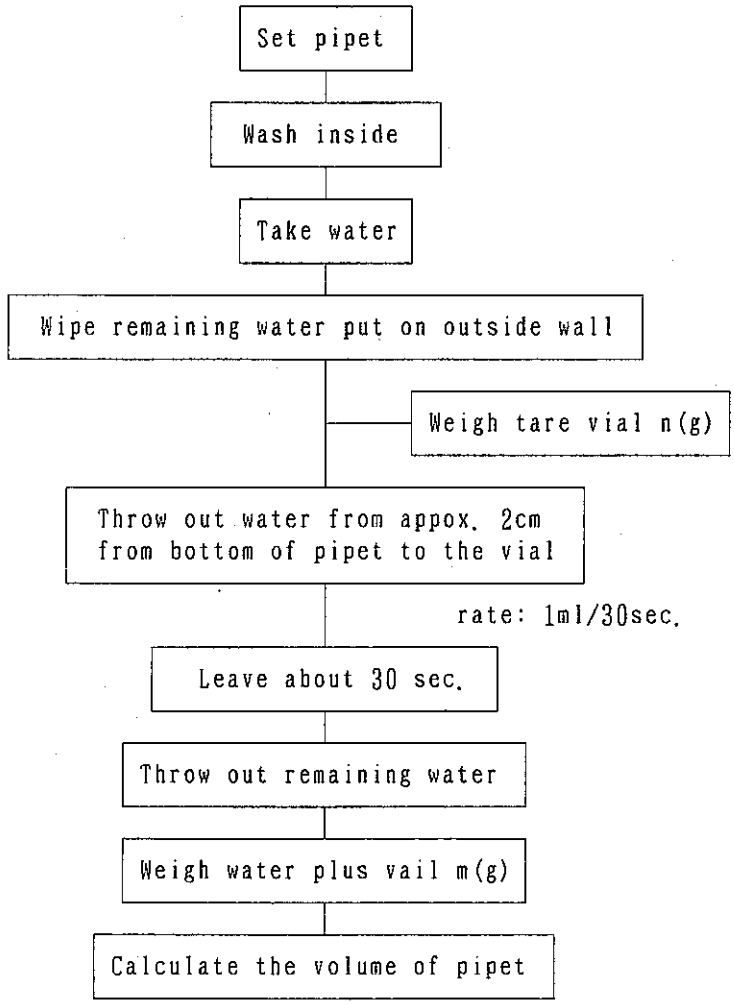
Volumetric sampling method, conventionally used, is based upon sample taking with a pipet, and dilution with a marked flask. Fig. 1 gives the procedure of pipet calibration, with an overflow type pipet and a marked flask are shown in Fig. 2 and Fig. 3 respectively. Since the volume-basd sampling relies on sampling with pipets and on dilution with measuring flasks, it tends to cause the following error problems:

1) Errors due to volume testing of the pipets and measuring flasks
2) Volume changes in the samples, vessels, pipets, flasks and vials due to temperature changes
3) Errors originating from the despersion in the amount of discharge due to the viscosity of the samples
4) Errors stemming from the dispersion caused at the time of marked-line adjustment for the pipets and measuring flasks

Besides the above, errors may occur because of the difference between the temperature at which the volume of the PNC's accounting vessel is measured (by the inspectors) and that at which the samples are treated on a volume basis in the PNC's Analytical Laboratory.

If the samples are treated and measured on a weight basis, however, no adverse effect can be observed principally. Namely, the value of density can be measured in the accountability vessel using manometers at the same time when volume is measured. Therefore nuclear material concentrations can be calculated without any effects of temperature. Bacause items depending on temperature can be compensated by using both density and volume at vessel, and gravimetric units on sample concentration. Further, since no proficient technique such as marked-line adjustment is required, the personal errors between operators will not occur. By using this system, the measuring errors of the volume- and weight-base methods were calculated for the actual plutonium product solutions produced in the Reprocessing Plant.

Gravimetric sample preparation can be easily automated because of its simple operation. The accuracy of gravimetric method depends on the reliability of balance so that it is important to keep balance always in fine condition.



Measure the temperature (T℃) and correct volume with appropriate density

$$V\ (ml) = \frac{m\ (g) - n\ (g)}{\rho\ (g/ml\ at\ T℃)}$$

Repeat three times on each pipet

Fig. 1 Procedure of over-flow type pipet calibration

Fig. 2 Over flow type pipet

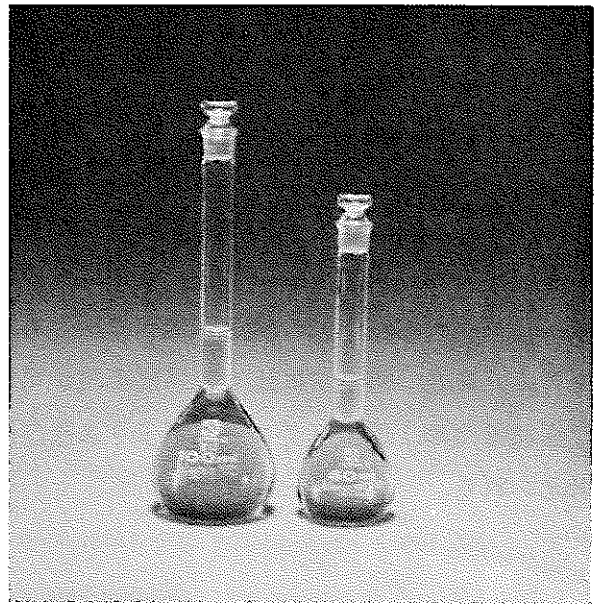


Fig. 3 Marked flask

3. DESIGN OF AUTOMATED GRAVIMETRIC SAMPLE PREPARATION SYSTEM

3.1 DESIGN CONCEPT

(1) Summary

Following points should be noted on the design of the automated gravimetric sample preparation system.

1. To accomplish more accurate measurement than volumetric method.
2. To use acid corrosion-resistant materials.
3. To install in the existing glove box.
4. Easy to maintain / repair through a glove.
5. Easy to control for inspectors of Japan government and IAEA.

Fig. 4 shows the scheme of sample preparation and analysis of the plutonium products at TRP[2)]. Plutonium product sample is taken for accountability analysis and inspection. Sample preparation consists of first sampling, dilution and second sampling. At the reprocessing plant, samples prepared are analyzed with controlled potential coulometry and mass spectrometry for plutonium concentration and isotopic composition respectively. On the other hand, samples for safeguards inspection are sent to IAEA SAL * and NMCC SAL**.

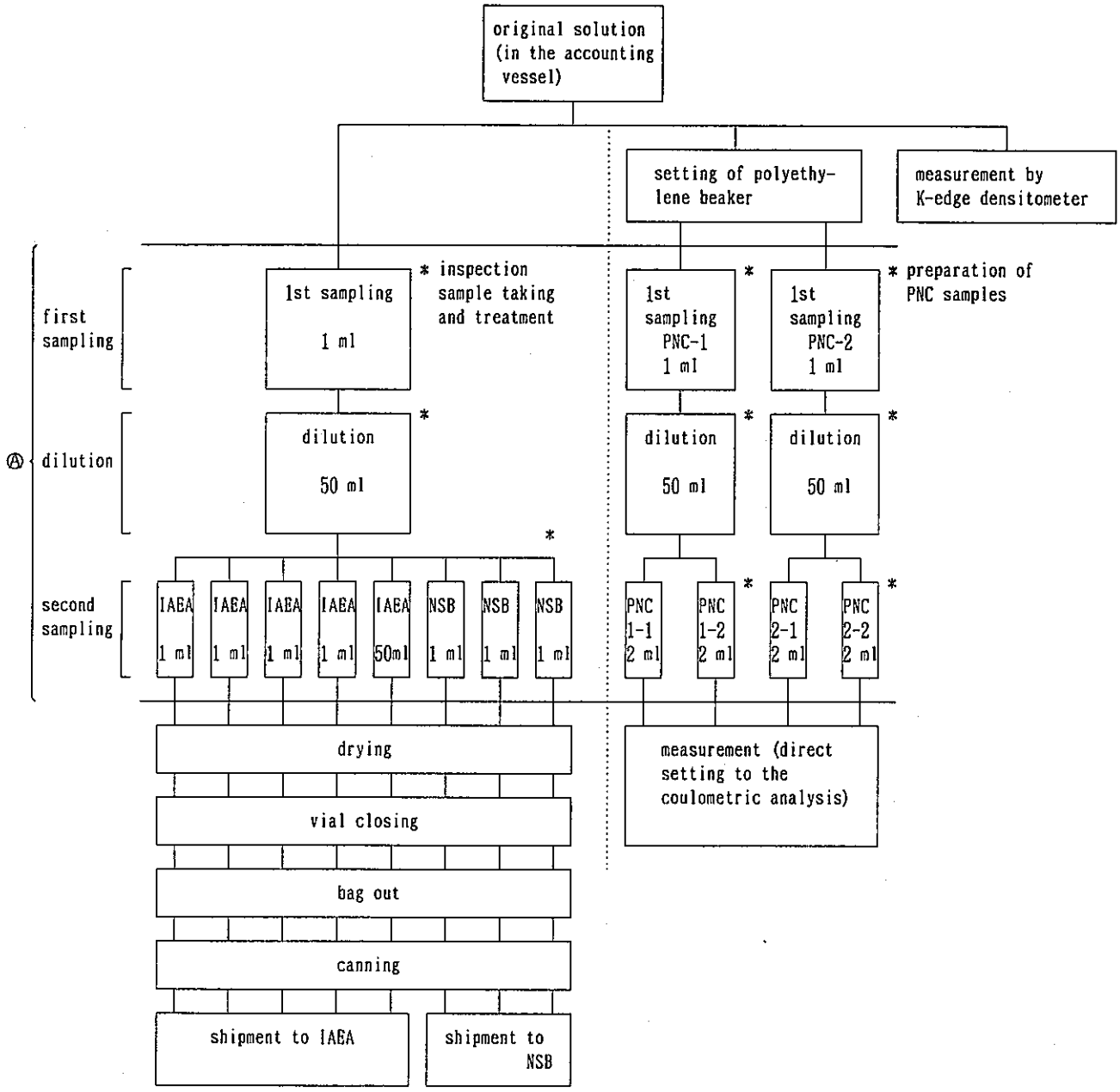
---

* International Atomic Energy Agency Safeguard Analytical Laboratory

** Nuclear Material Control Center Safeguard Analytical Laboratory

Besides sample taking for safeguards inspection, K-edge densitometry, one of the sophisticated non destruction analysis method has been introduced to TRP in order to obtain results rapidly. These inspection samples are prepared in presence of inspectors of Japan government and IAEA (Fig. 5). Inspectors observe the sample preparation to confirm if the operation goes on properly.

Volumetric method used to be adopted in the portion of Fig. 4 A. These operations have been changed to gravimetric method. All operations but setting tare vials are automatically performed with a robot arm, an diluter and a balabce.



* The volumes written under the terms "sampling" and "dilution are rough, and their correct values are determined by weighing.
IAEA: Samples for international inspection analysis
NSB : Samples for national inspection analysis
PNC : Samples to be analysed by the Tokai Reprocessing Plant, PNC

Fig. 4 Plutonium product sample preparation

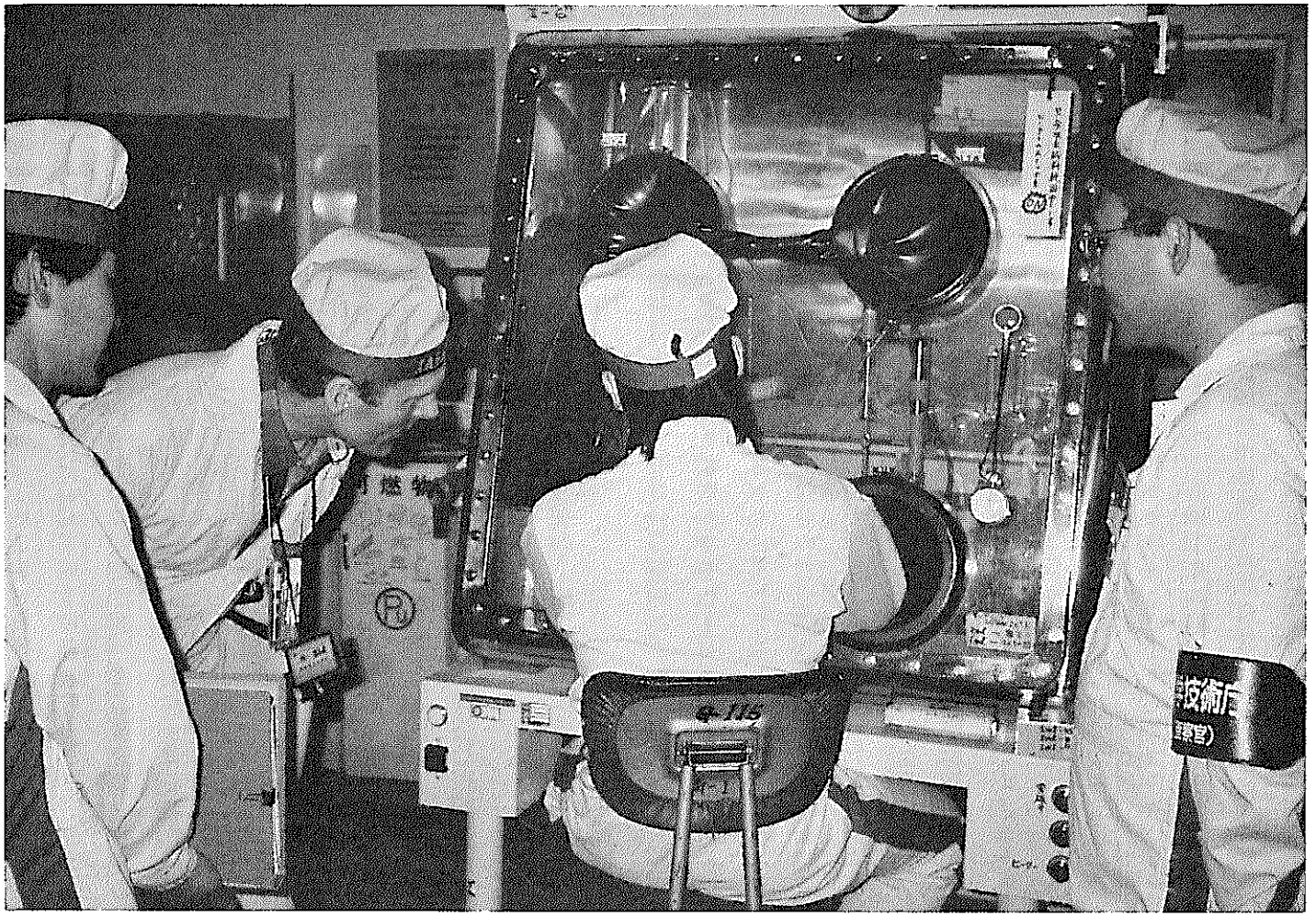


Fig. 5 Preparation and inspection of plutonium product sample

(2) Several studies for designing of the system

Plutonium samples are dealt with in the glove boxes for accountability analysis at TRP. These glove boxes are divided into two groups called, line I and line II. Sample preparation and measurement are carried out in line II (Fig. 6).

The automated gravimetric sample preparation system is also installed in line II (Fig. 7). The glove box for automated gravimetric sample preparation system made of stainless steel has larger volume than that of normal type PVC glove box in consideration of systems maintenance (Cubic type glove box, the floor area: 1m × 1m). Each box has vinyl bag ports of 30cm ϕ or 20cmϕ to brig materials into the box. Any materials necessary to put into glove box must be smaller than vinyl bag port, whereas more than half of floor area should not be occupied by the installed system from the view point of spare room for other operations and maintenance.

It seemed to be essential to introduce robot arm to solve the space problem shown above. There were two concepts of setting robot arm (Fig. 8).

1) Robot arm is installed in glove box
2) Main part of robot arm except grip hand is installed at outside of glove box.

Main part and grip hand of robot are connected with the boot.

The letter concept suits the case that there is enough room in glovebox to install a large system such as commercially available robot. In this case main part of robot arm is set outside of glove box. This application enable to prevent corrosion on the robot arm materials. It is essential to protect the connection from air leak. That requires special device on the connection and boot. From the reasons shown above, it was concluded that the system was difficult to install in the existing PVC glove box.

Former concept[1] seems to suit the idea to adopt small instrument, while corrosion problem must be overcome. Based on above studies, formet concept was chosen for system designing.

TRP has developed a miniature robot arm because of no commercially available robot at the time.

Serbo motor can be regarded the most suitable way of controlling robot arm.

Serbo motor robot, however, needs large room in general. In conclusion pulse motor was adopted for robot arm designing because pulse motor can be small and can control robot arm operation accurately. In this case, pulse motor type robot arm has to be operated as sequential operations by programming in comptuter prior to routine use. In addition, balance, cups and other parts must be set in accurate position. It appears to be considered a defect. However, there is no requirement to operate grip hand so exactly on this sample preparation procedure. Furthermore, all parts are made of acid proof materials such as stainless steel, PVC and teflon so as not to be corroded in a glove box.

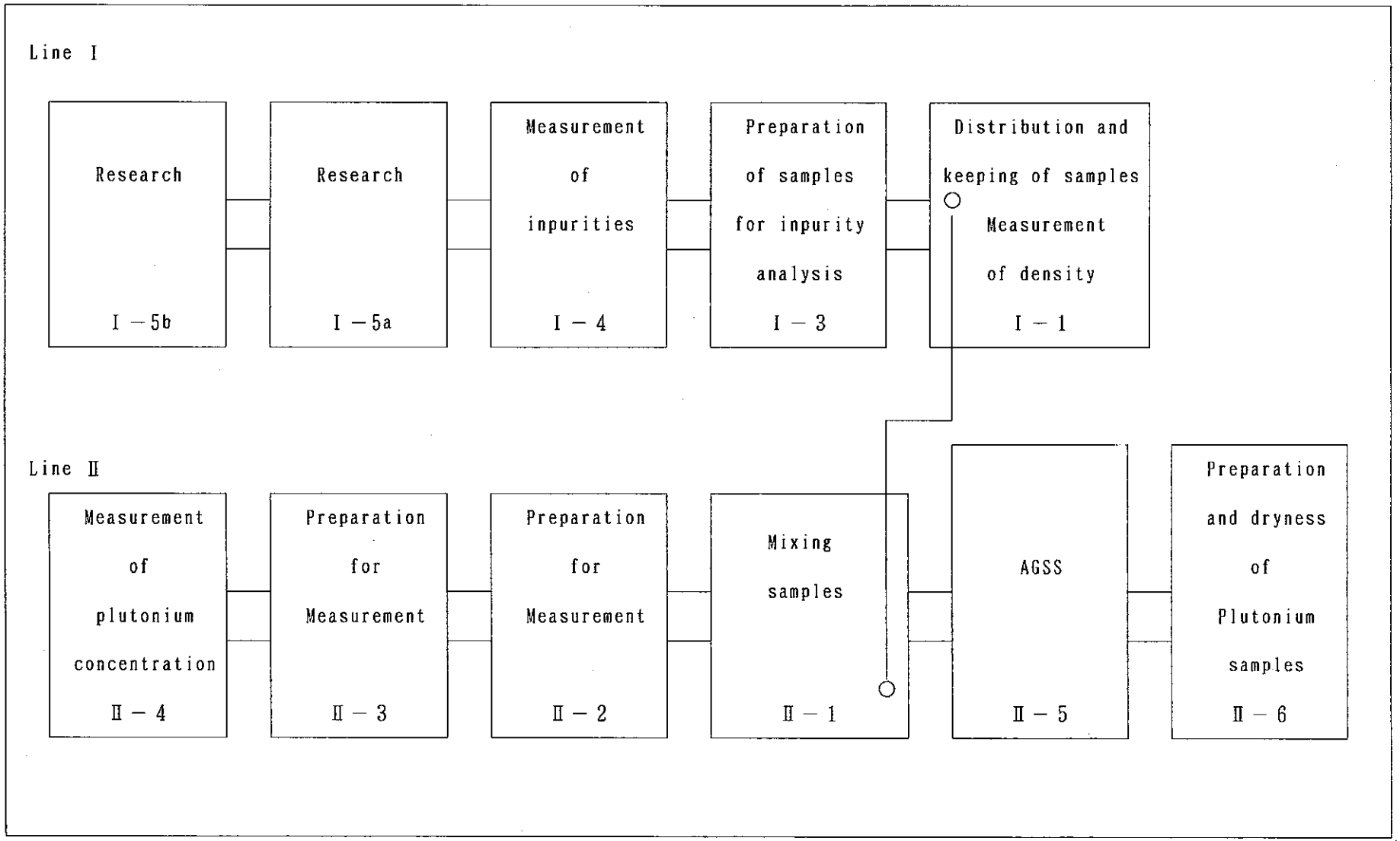


Fig. 6 Layout of accountability analysis glove box lines

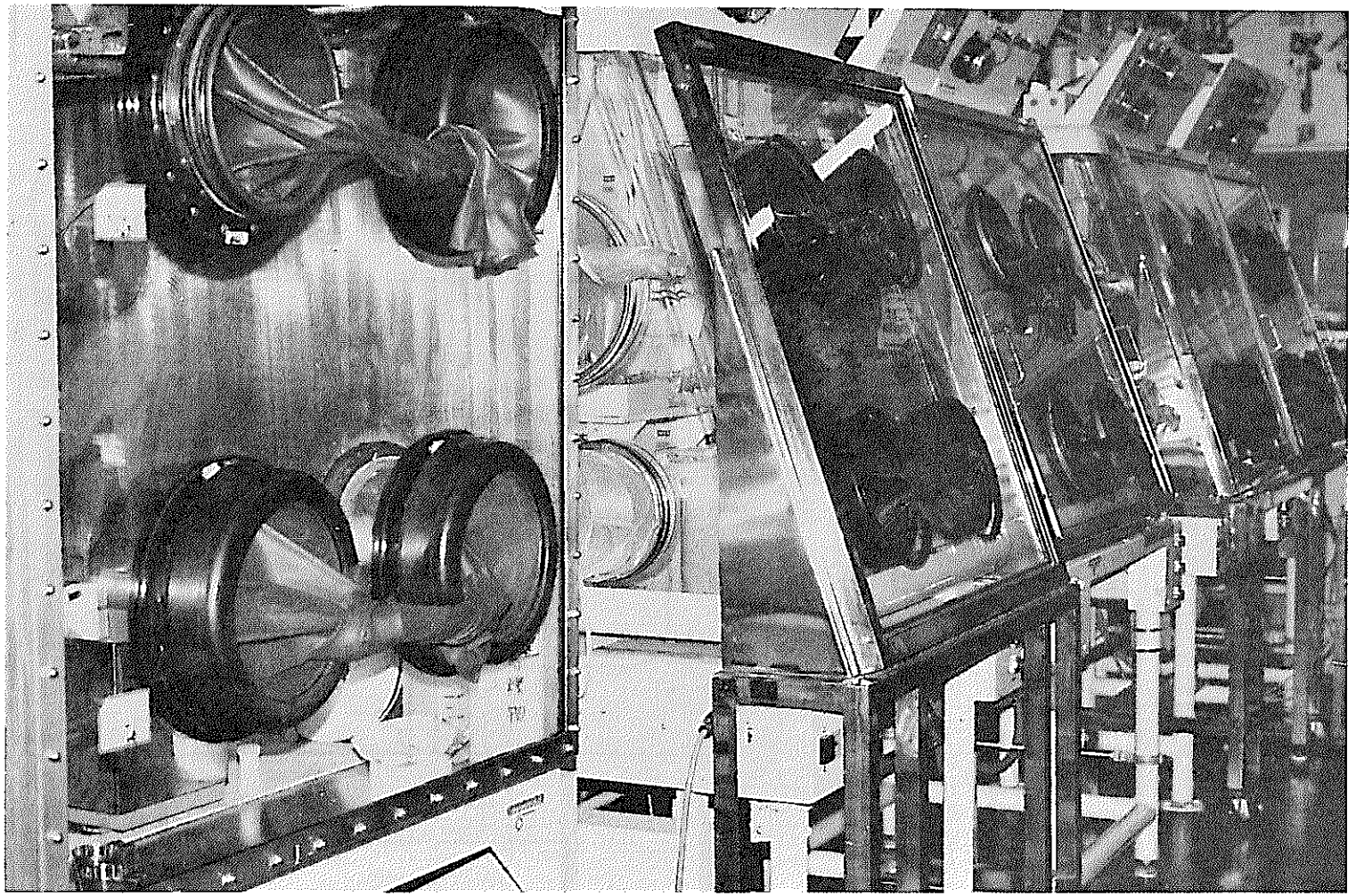
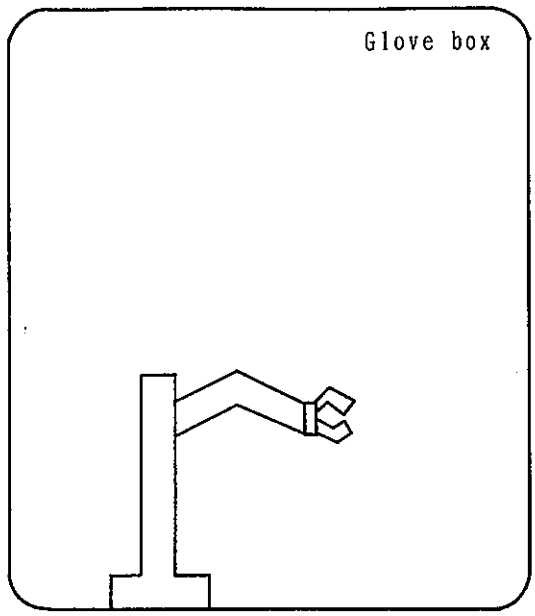
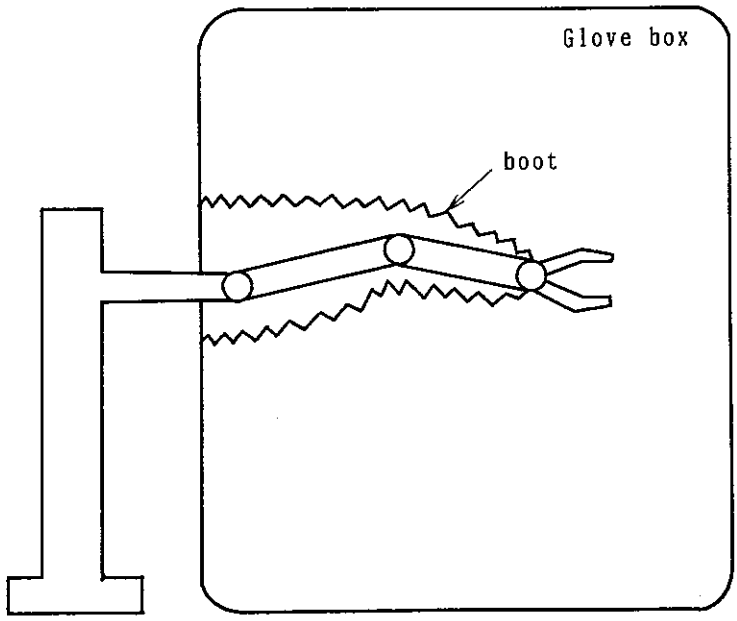


Fig. 7 Glove box line for accountability analysis
(AGSS is installed in the left side box)



1. Robot arm installed in blove box



2. Robot arm isolated from inside glove box

Fig. 8 Design concepts of robot arm

3.2 SYSTEM DESCRIPTION

The system consists of the mechanical and control subsystems; the former includes an electric balance, robot arms and a turn table, whereas the latter a computer and an interface.

Fig. 9 shows appearance of automated gravimetric sample preparation system, whereas Fig. 10 gives block diagram of the system. Mechanical sybsystem has been installed in the glove box and operated by the control subsystem set at the outside of the glovebox. There are two turn-tables; one for setting dilution cups and another for setting vials. The control subsystem is composed of control panel, interface and display panel.

The mechanical subsystem make first sampling of stock solution, weighing samples, dilution, weighing diluted samples, second sampling and weighing secondary samples. All operations are commanded by control subsystem.

Fig. 11 to Fig. 13 show the photos of the external view of the system, the control subsystem and the mechanical subsystem. The mechanical and the control subsystem functions are discussed later in detail.

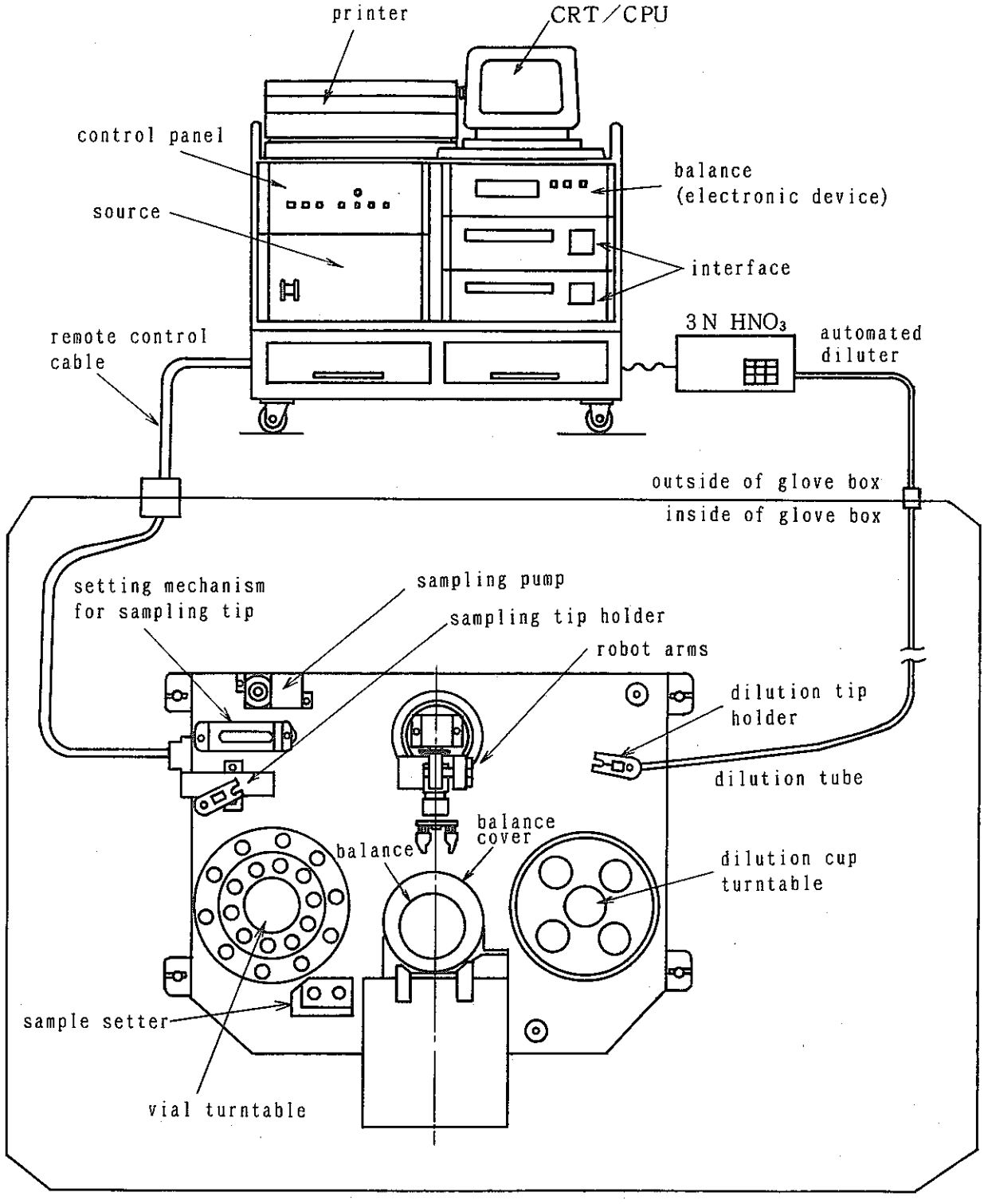


Fig. 9 External appearance of automated gravimetric sample preparation system

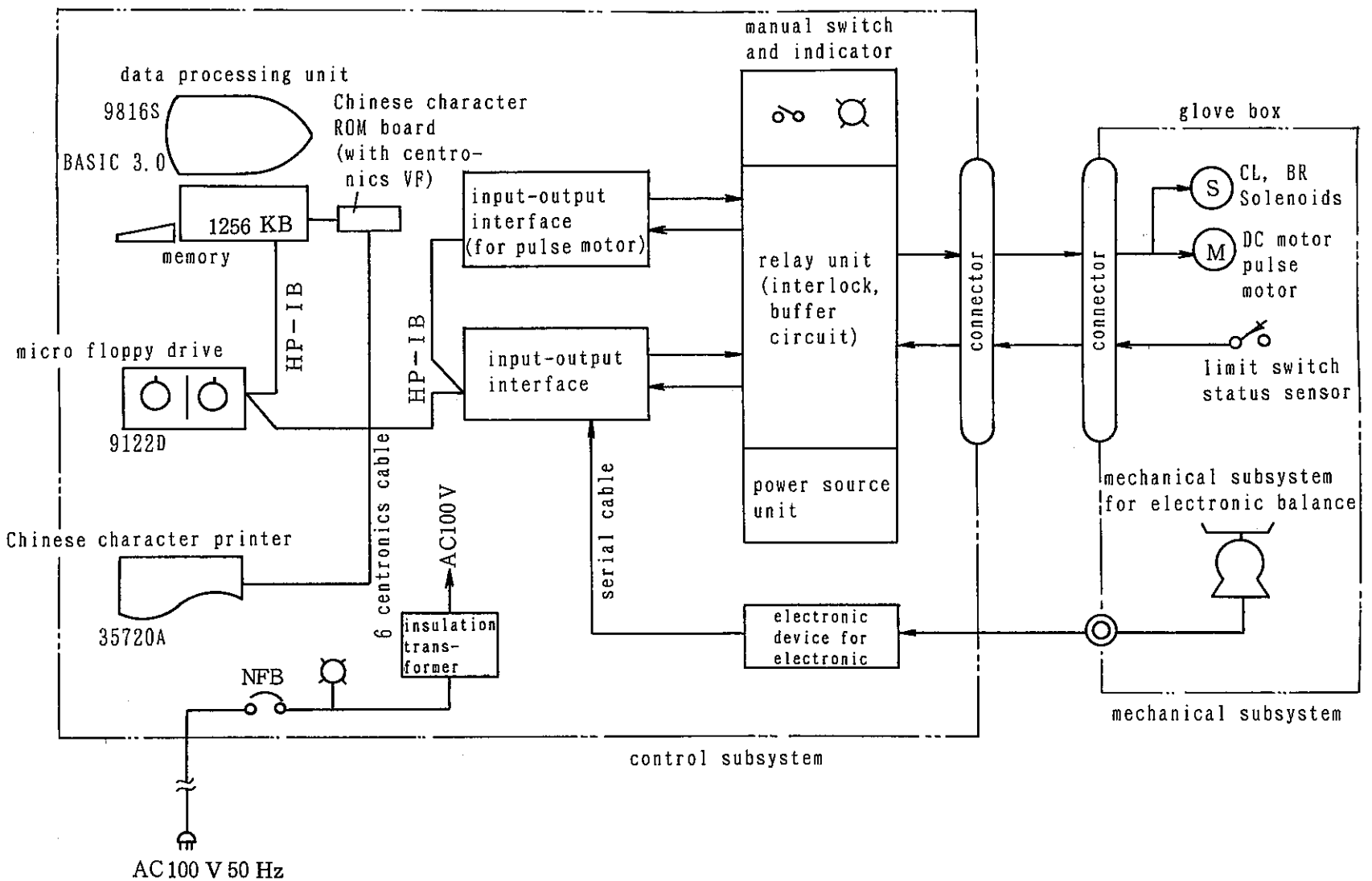


Fig. 10 Block diagram for automated gravimetric sample preparation system

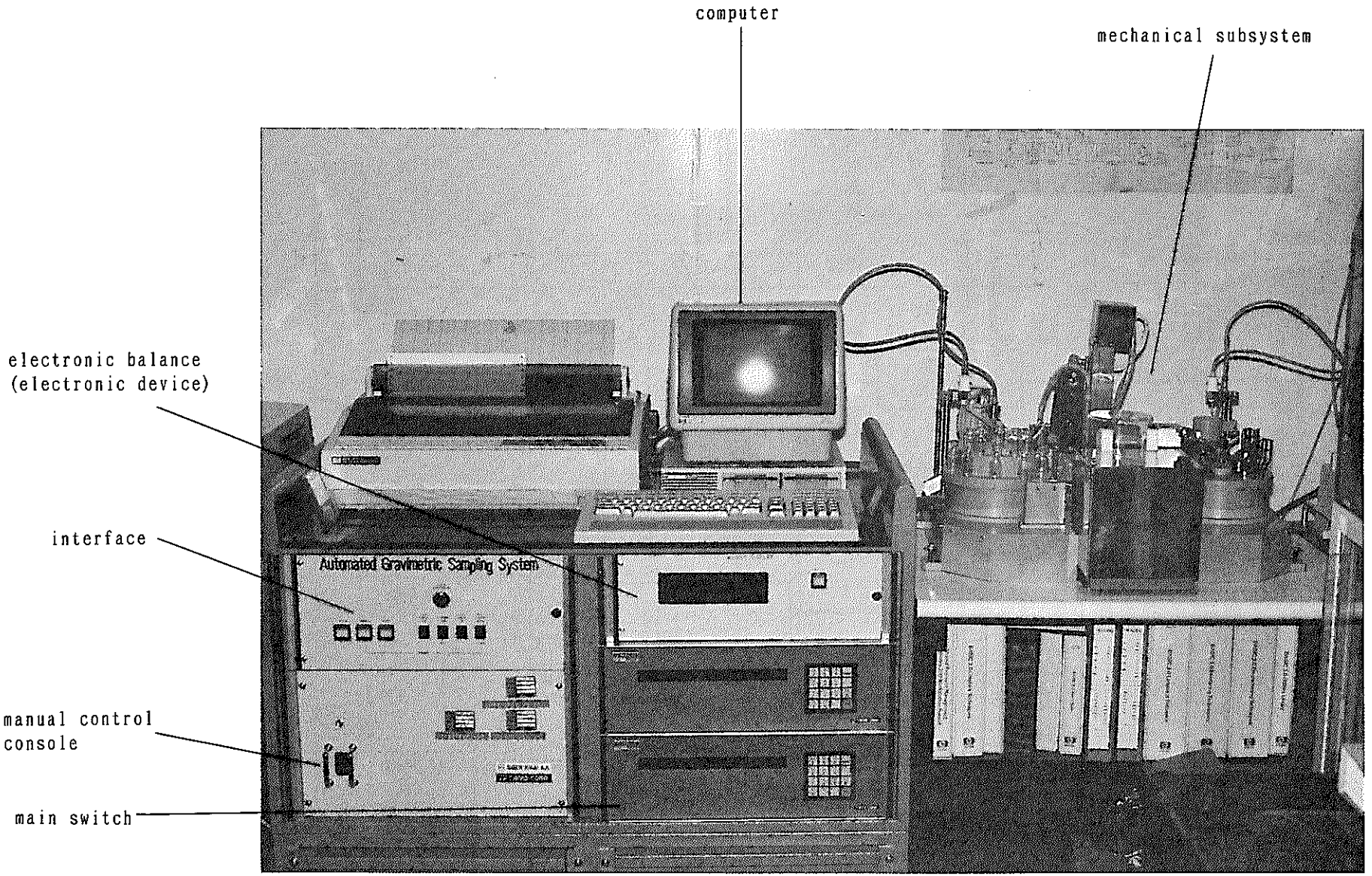


Fig. 11 Automated gravimetric sample preparation system as a whole

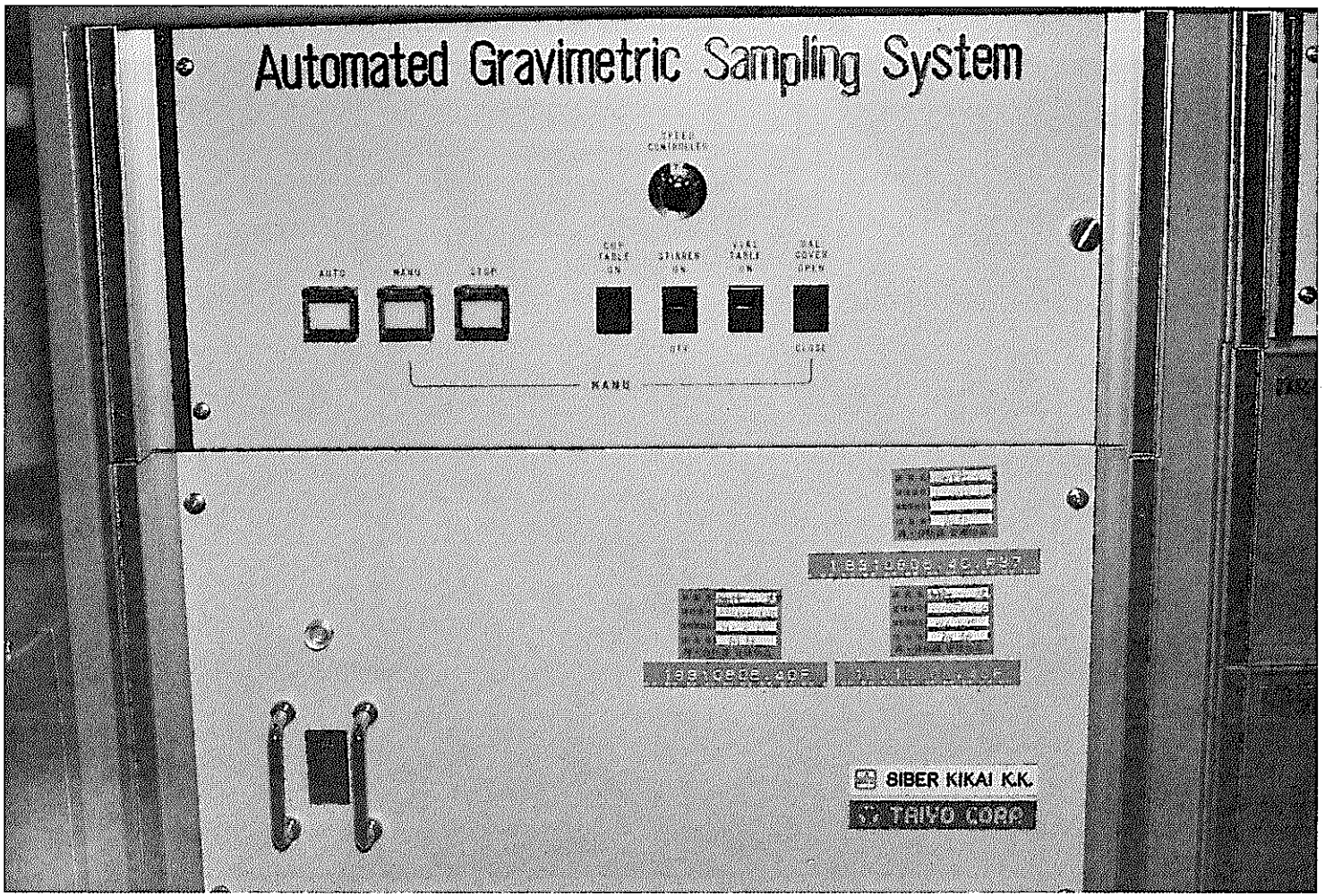


Fig. 12 Manual control console for control subsystem

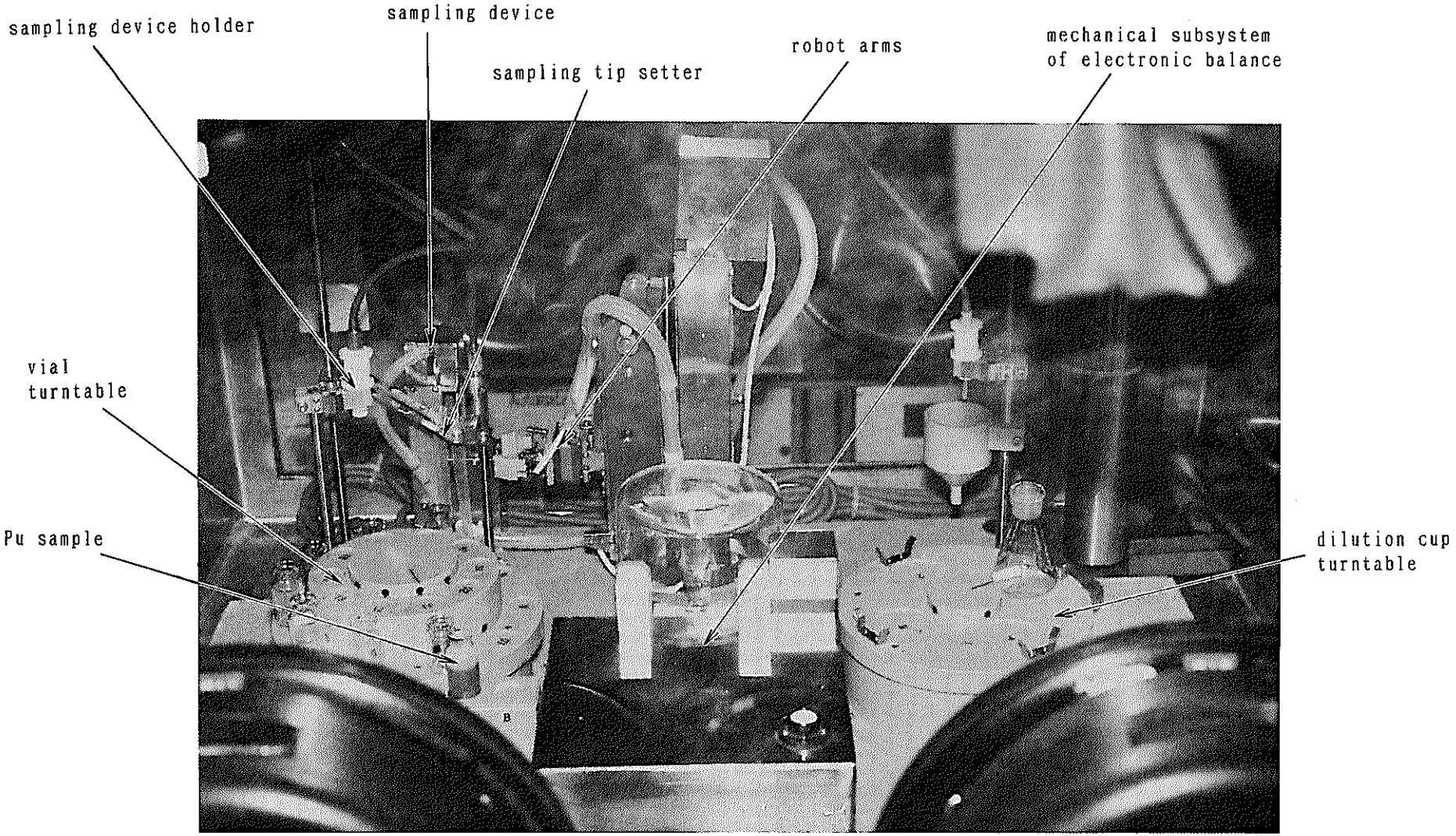


Fig. 13 Whole mechanical subsystem of automated gravimetric sample preparation system (installed in the glove box)

(1) Robot arm

The robot arm (Fig. 14) is capable of transferring cups, vials and tips connected to sampling device to assigned positions (Fig. 9). The small sized robot arm, 380mm height and 100mm wide, has been achieved by using pulse motor.

Robot arm moves up and down within 220mm from the origin (0-point) stretchs up to 250mm and rotates 180 degrees (Fig. 15). This robot arm is operated continuously by main program "SAMPLING", because every steps of pulse motor can be controlled by just CPU.

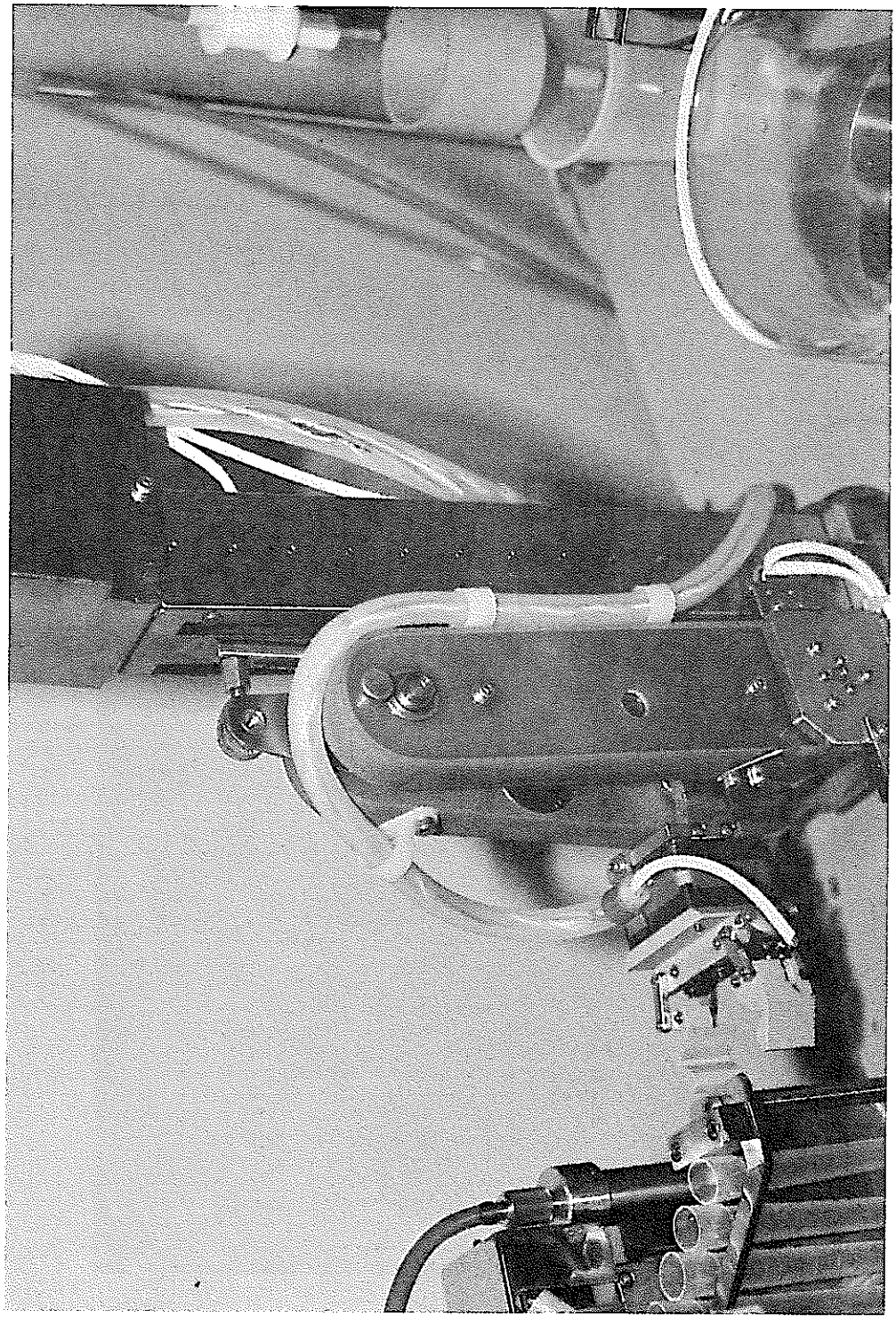


Fig. 14 Robot arm

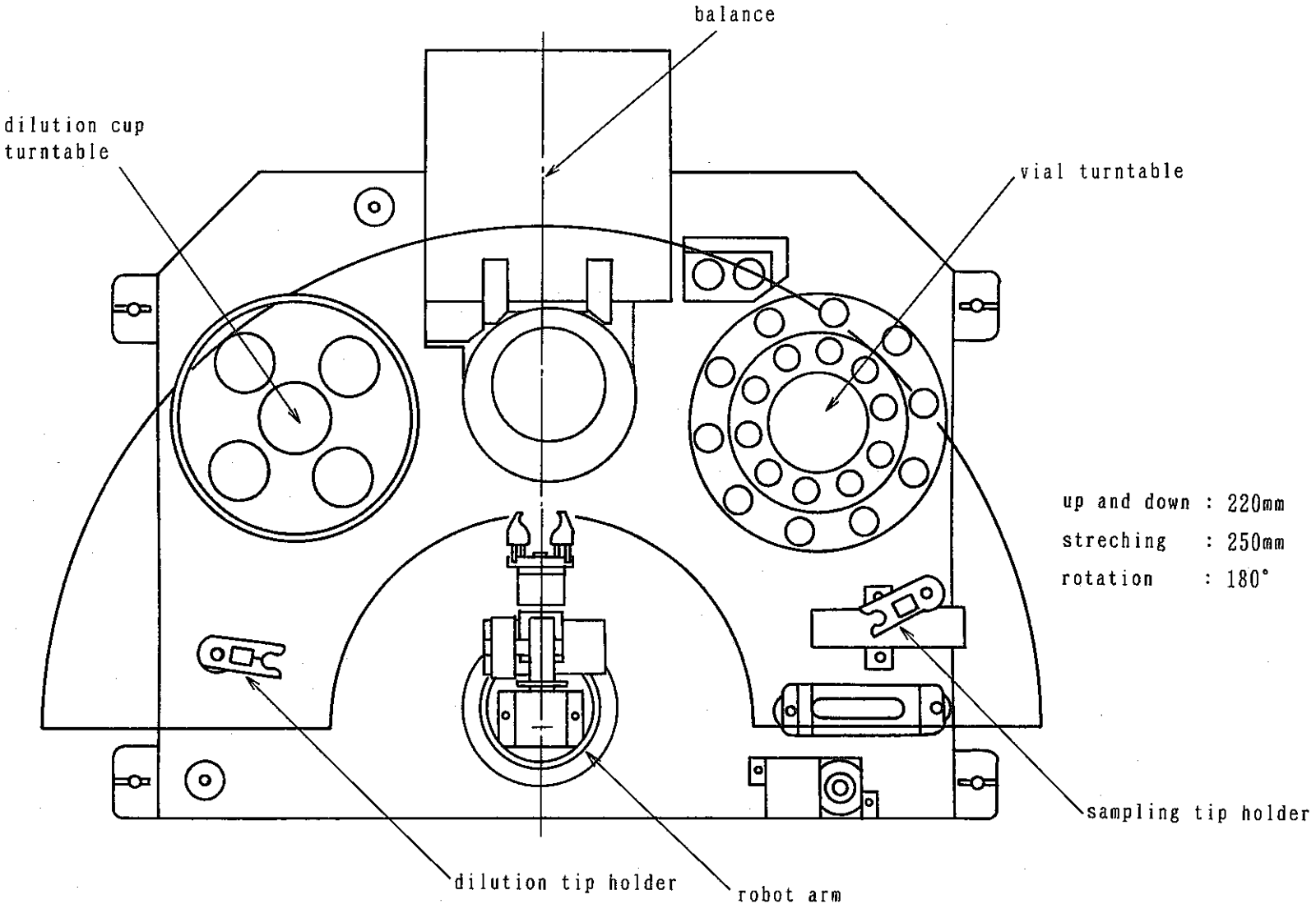


Fig. 15 Area where robot arm can be operated

(2) Balance

Balance is one of the most important parts of the system as well as the robot arm. Referring to the scheme of sample preparation on volumetric method, effective region of balance has been discussed.

① maximum value weighing

Though diluted volume is 50ml for routine preparation, it may be changed to 100ml some cases. Hence the maximum of weighing can be calculated as follows,

(Volume of diluted solution × density of diluted solution) + tare ≒ maximum weight

(100ml × 1.2g/cm³) + about 50g ≒ 170g

Taking account of margin of weighing region, 200g has been chosen as maximum effective value of weighing.

② The least figure of balance

The reporting result of accountability analysis requires four figures. Thus 0.1mg should be enough for the sample preparation scheme.

③ Composition

The balance is divided into two parts; electric and mechanical parts, installed in the glove box in consideration of corrosion. The balance, METTLER AE-200 (Fig. 16), has been satisfactorily chosen based upon conditions mentioned above except that mechanical part must be separated from electric one including print board for calculation and control.

Fig. 17 shows the way of separation and modification, while of the balance. Fig. 18 gives the view of the miniaturization of the balance on height. The schematic of automatically controlled balance cover is shown in Fig. 19, whereas overview of the modified balance is given in Fig. 20.

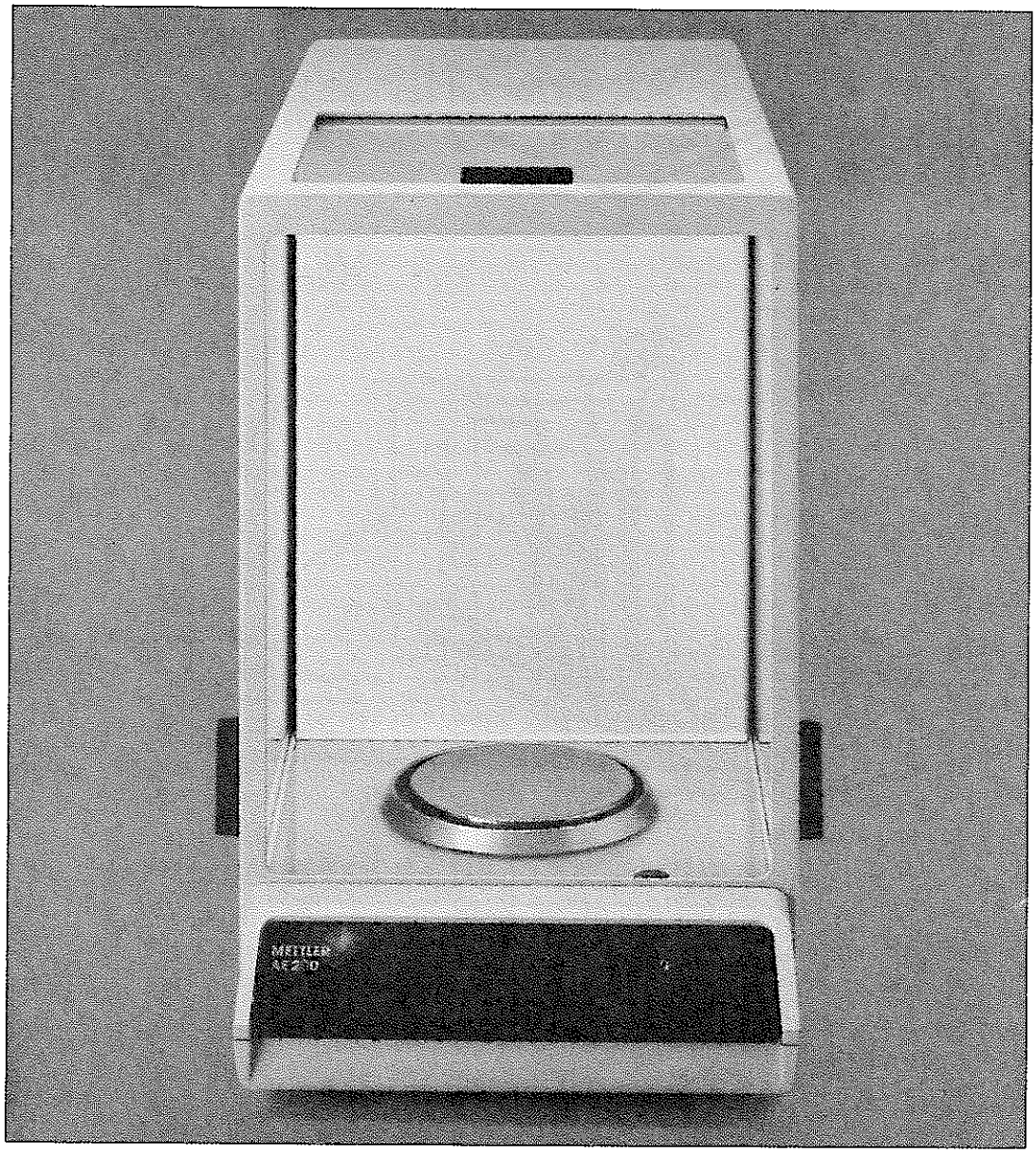


Fig. 16 Balance (AE-200)

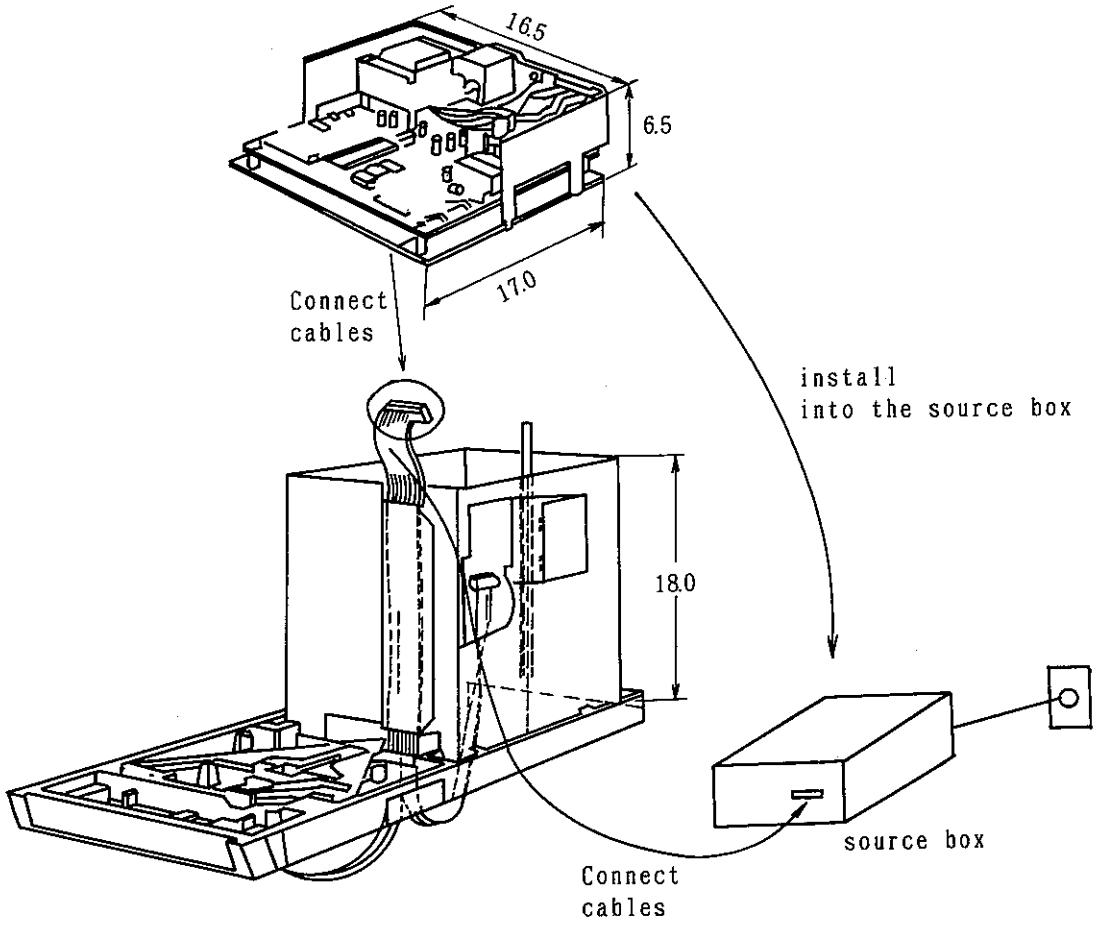


Fig. 17 Separation and modification of the balance, AE-200

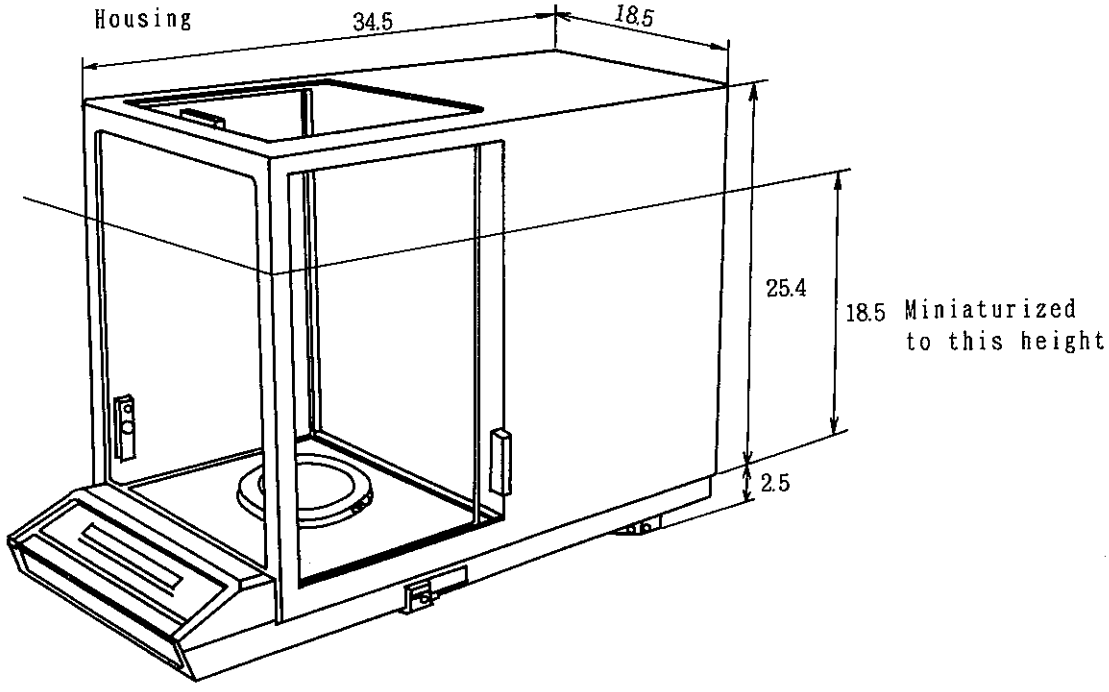


Fig. 18 Miniaturized balance, AE-200

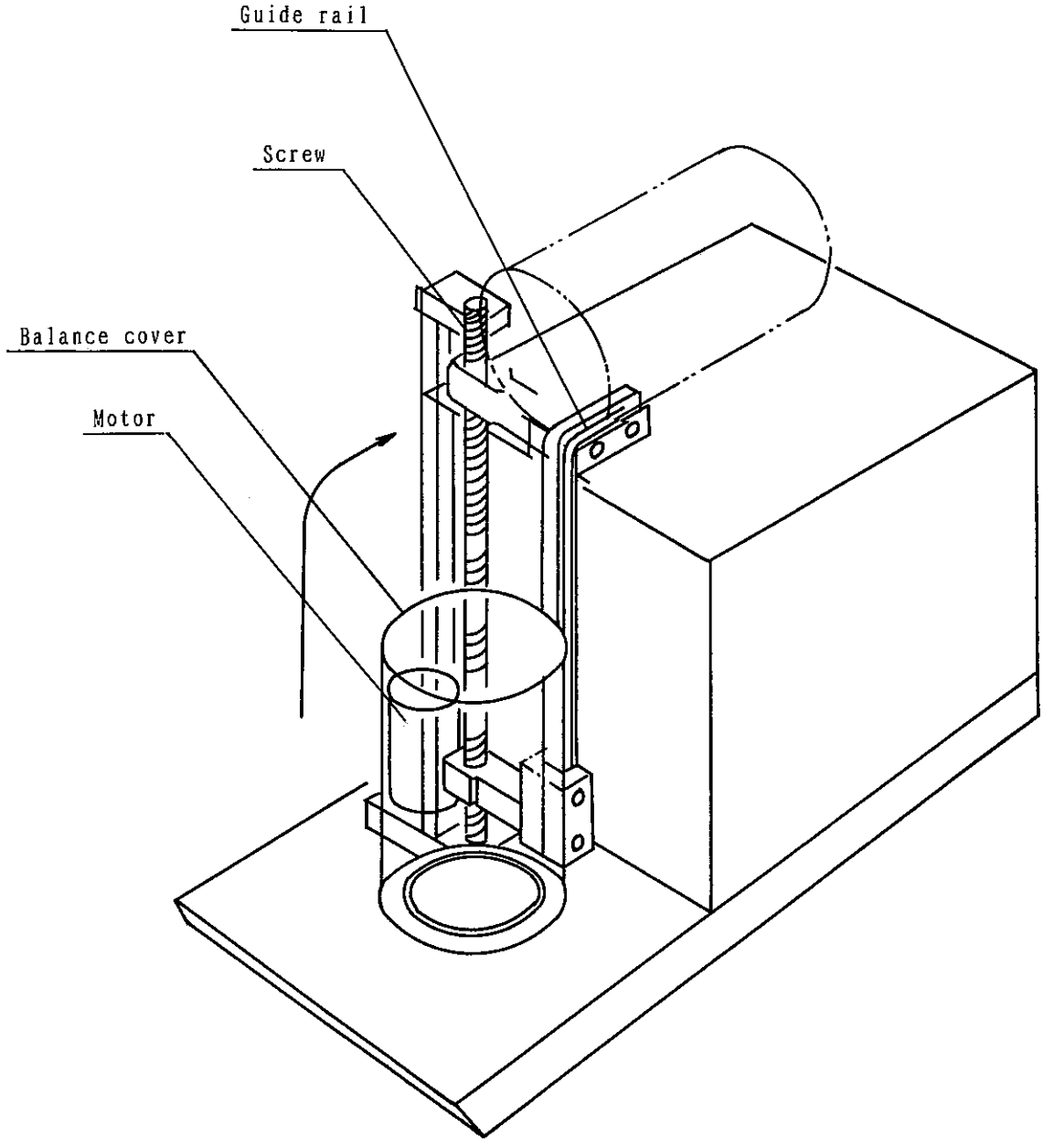


Fig. 19 Automatically controlled balance cover for AE-200

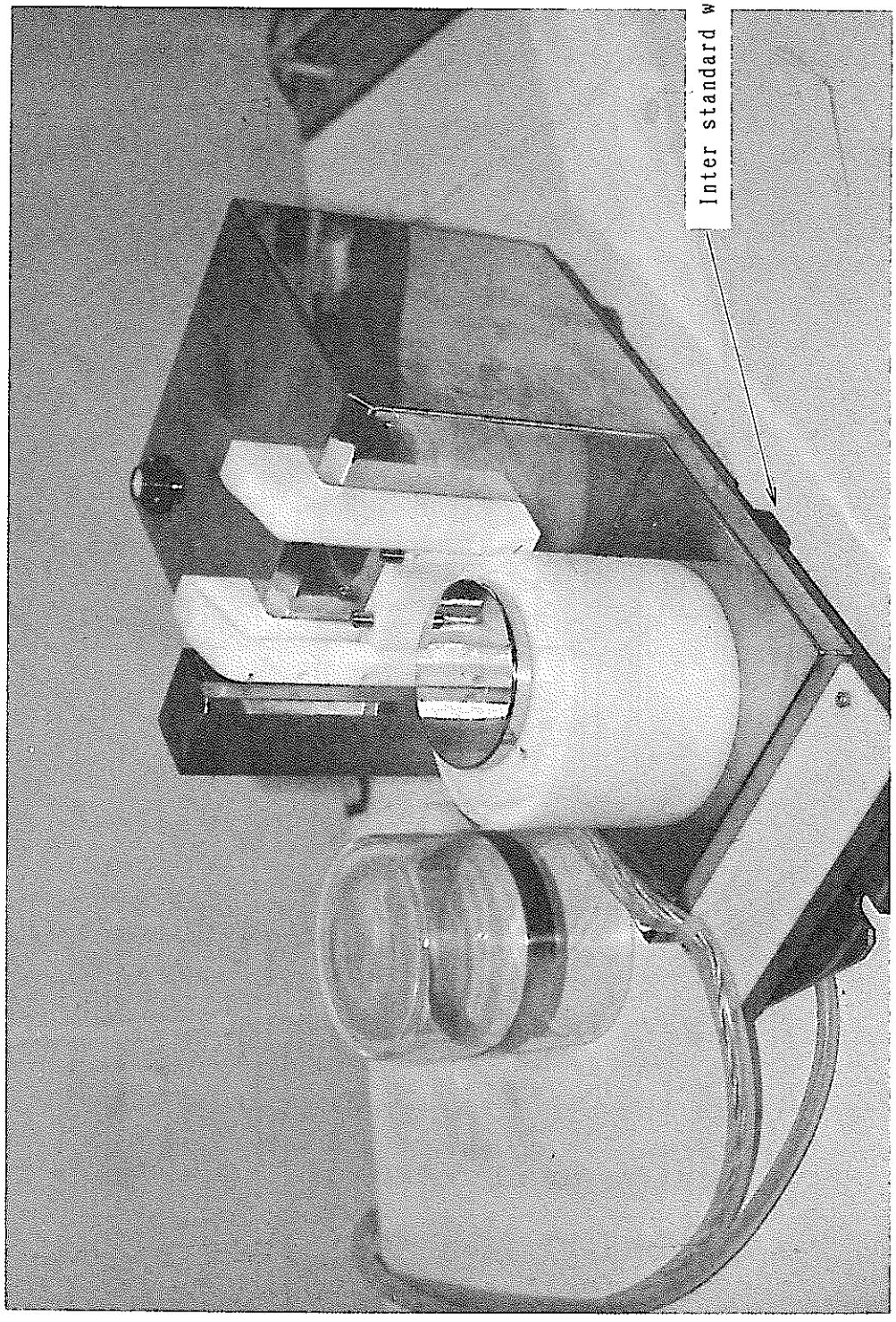


Fig. 20 Electric balance

(3) Turntables

There are two turntables, to put dilution flasks, vials and up to four flasks for dilution can be set on the turntable for diluting samples. The turntable for vials has 2 rounds, which is capable of setting up to twenty vials for DA samples (Refer to Fig. 21). The diluted samples for TRP accountability analysis is set on the other exclusive turntable. This exclusive table can be also used to take sample solution into coulometry polyethylene cup directly. Then plutonium contained in the solutions in the coulometry polythylene cups are measured by the controlled potential coulometry for the facility side accountability.

(4) Automated diluter unit

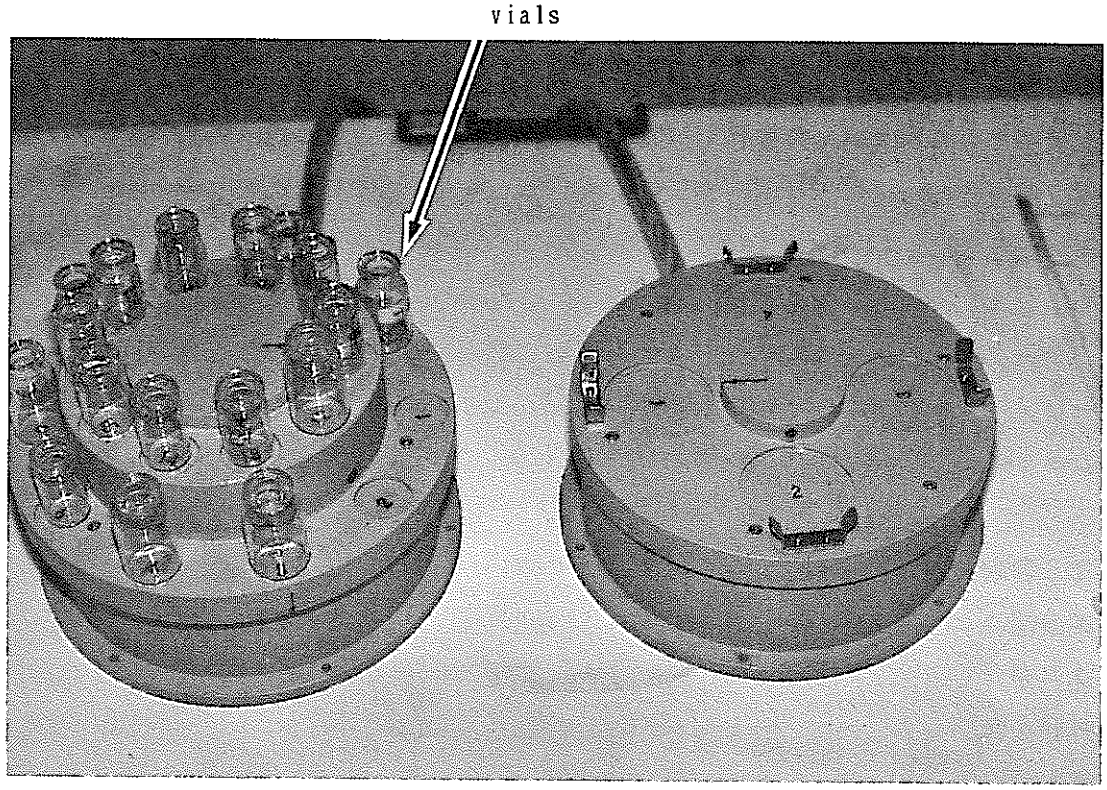
The injector of the automated diluter is connected with the diluter by a tube to introduce solution into the inside of glove box (Fig. 9). The automated diluter, Mettler type DL-20, and the injector are shown Fig. 23 and Fig. 24.

3N of nitric acid is used for the dilution. All operators have to do is to put the data of dilution volume into the computer.

(5) Sampling unit

The sampling unit is used to take aliquots of the accountability sample of plutonium product, as a first sampling. It consists of sampling head, including tip and sampler with volumetric cylinder. Those are connected each other by a tube.

Maximum sampling volume is 4.5ml. Fig. 25 shows the sequential movement of setting the tips by the robot arm. The view of the 1st sampling is shown in Fig. 26 and 27.



(for DA sample) (for dilution)

Fig. 21 Turntable for vials and dilution cups used for safeguards

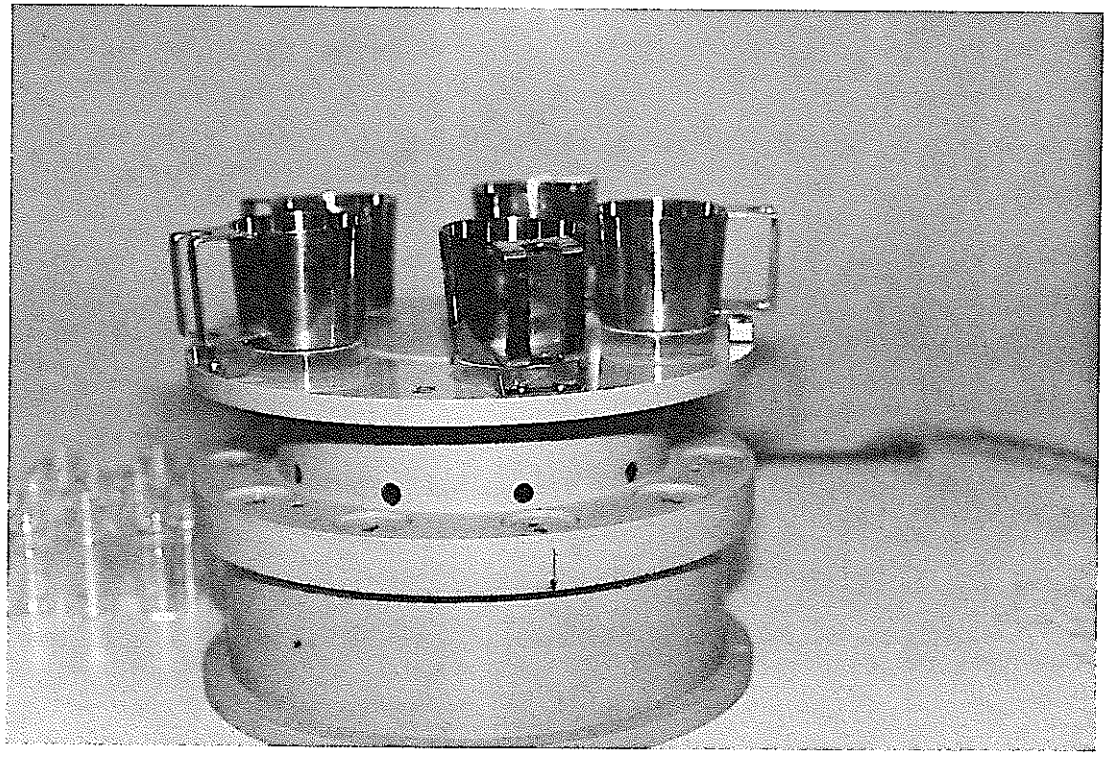


Fig. 22 Annex turntable used for TRP accountancy

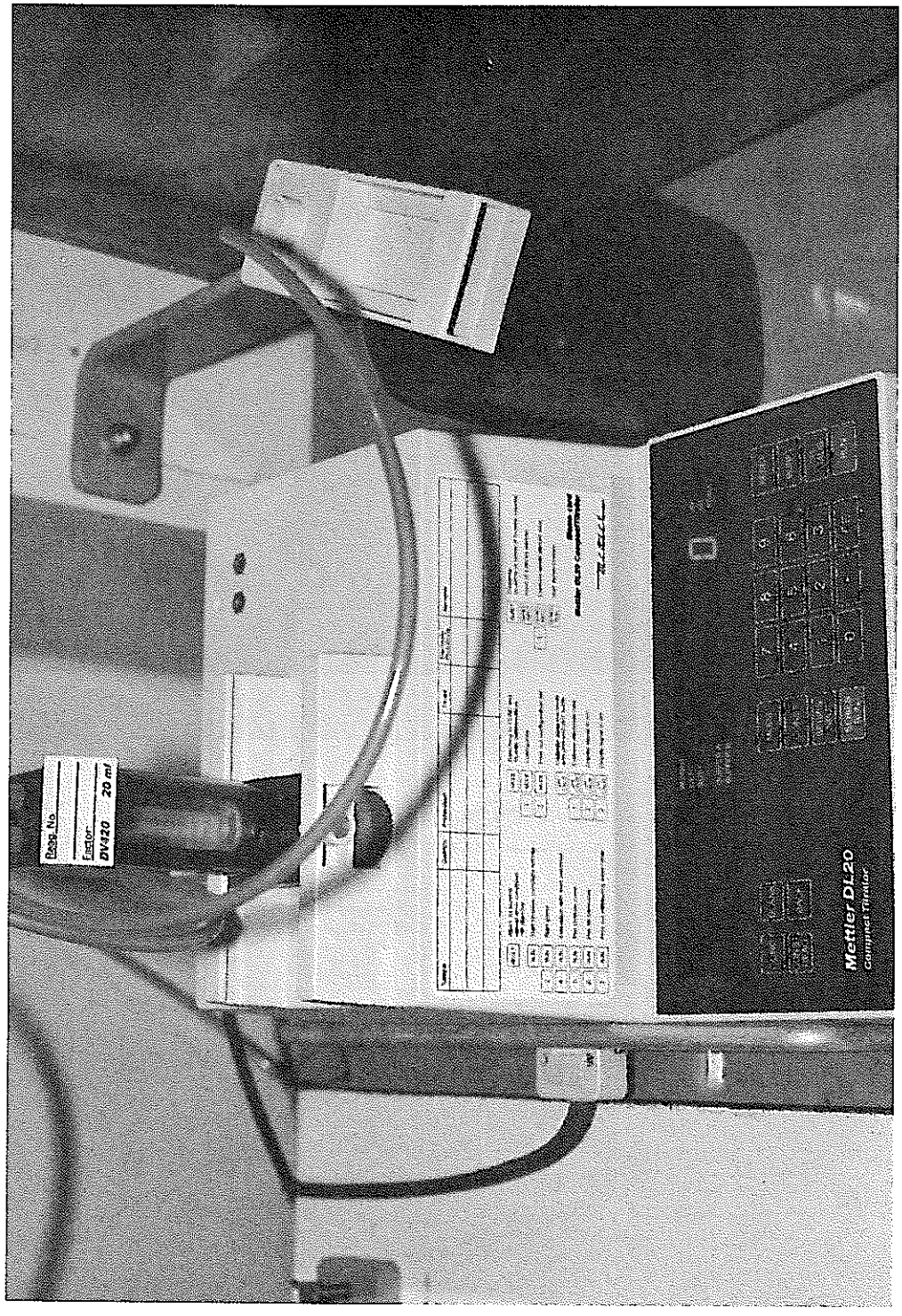


Fig. 23 Automated diluter

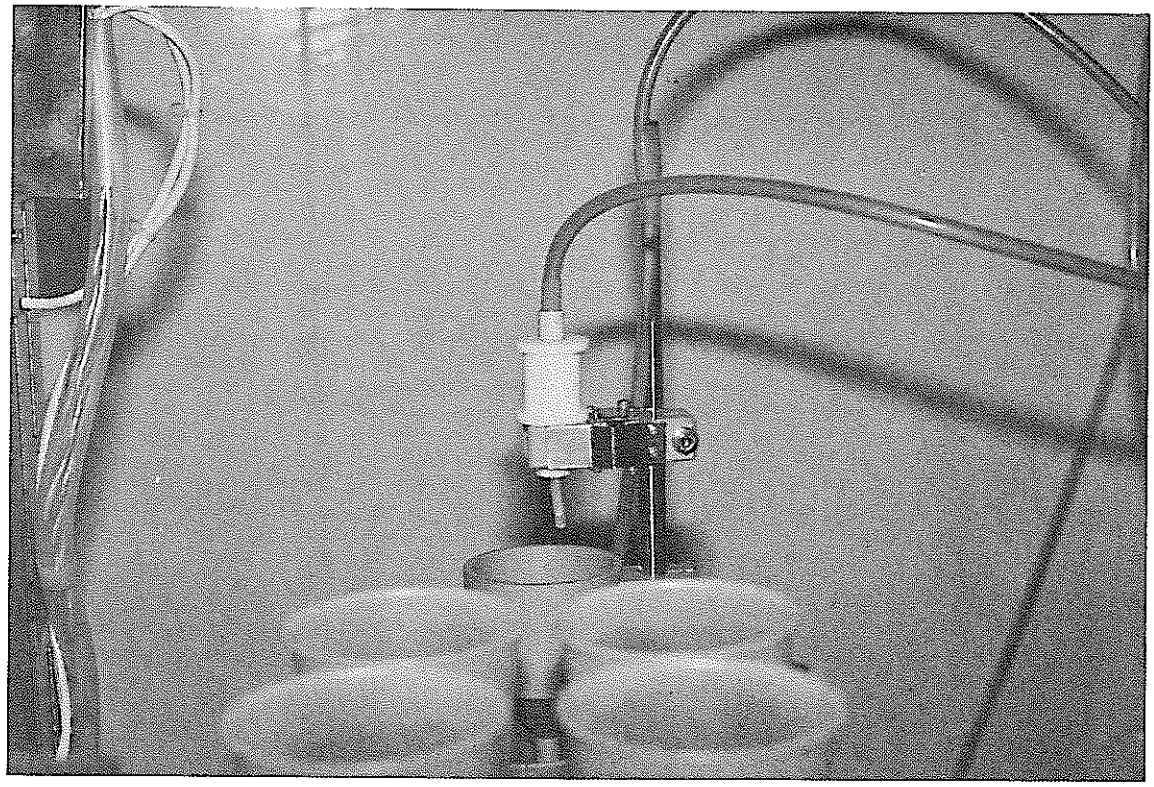


Fig. 24 Injector to be operated with a robot arm

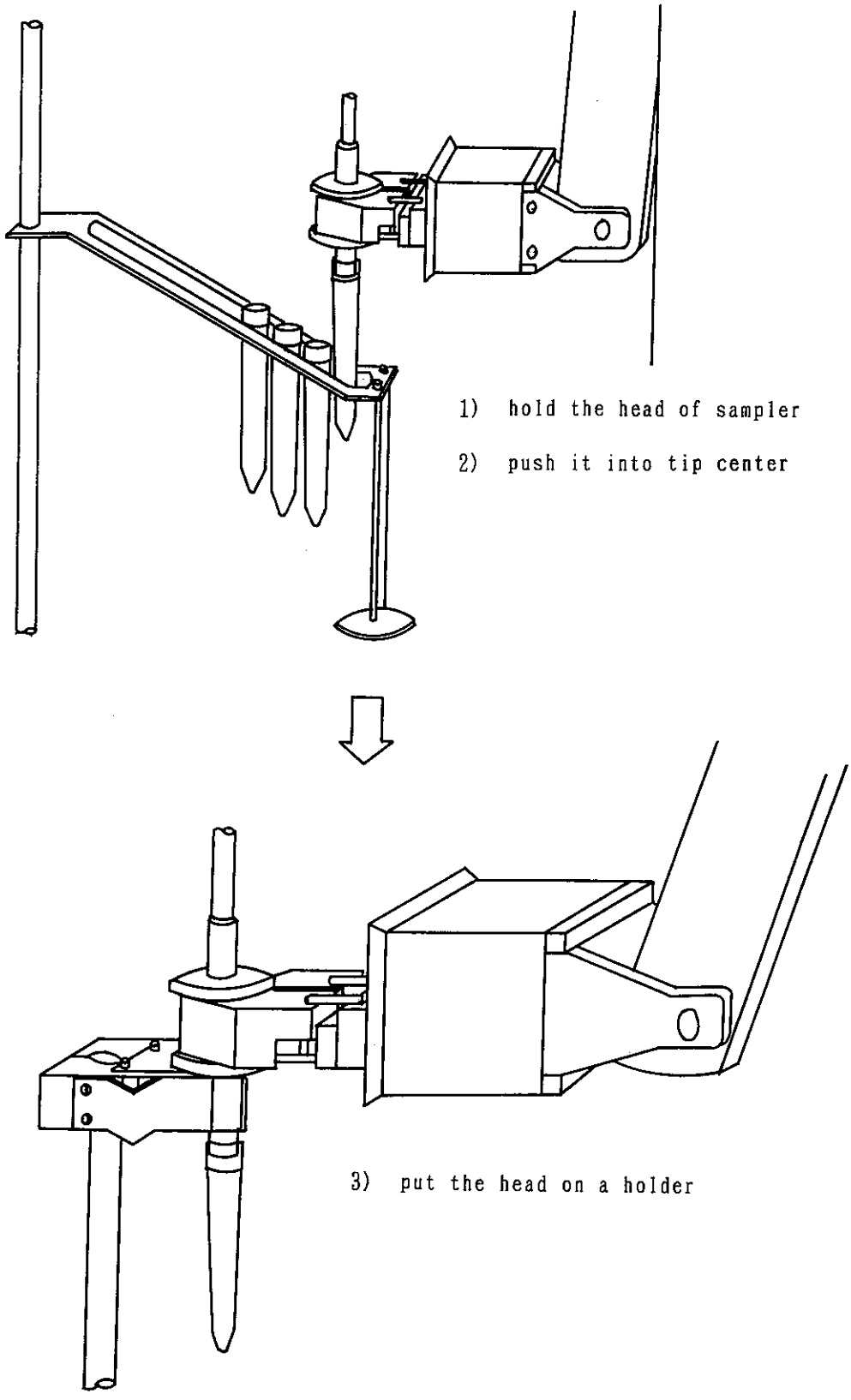


Fig. 25 Sampling tip with a robot arm

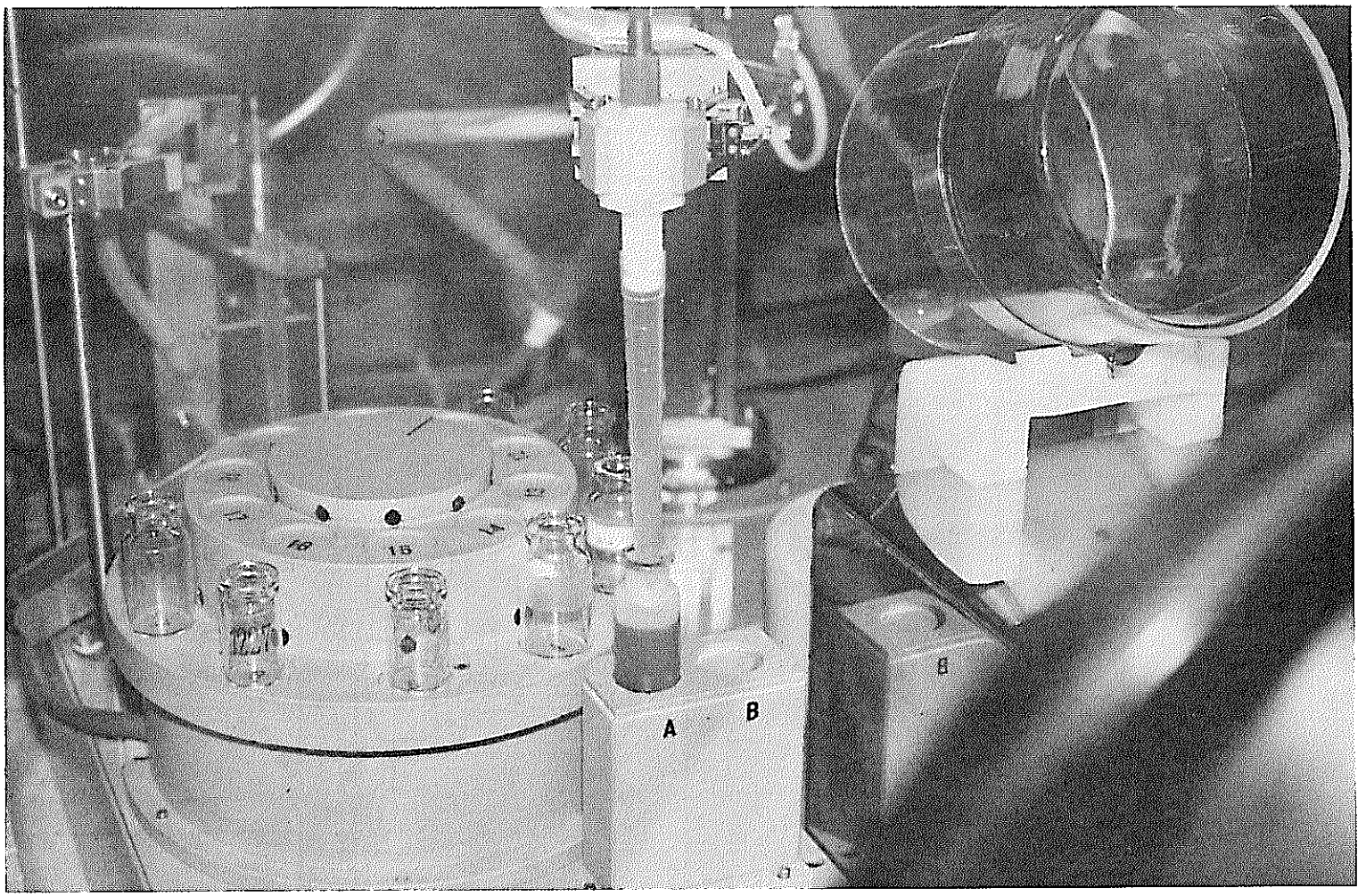


Fig. 26 Primary sampling of plutonium solution

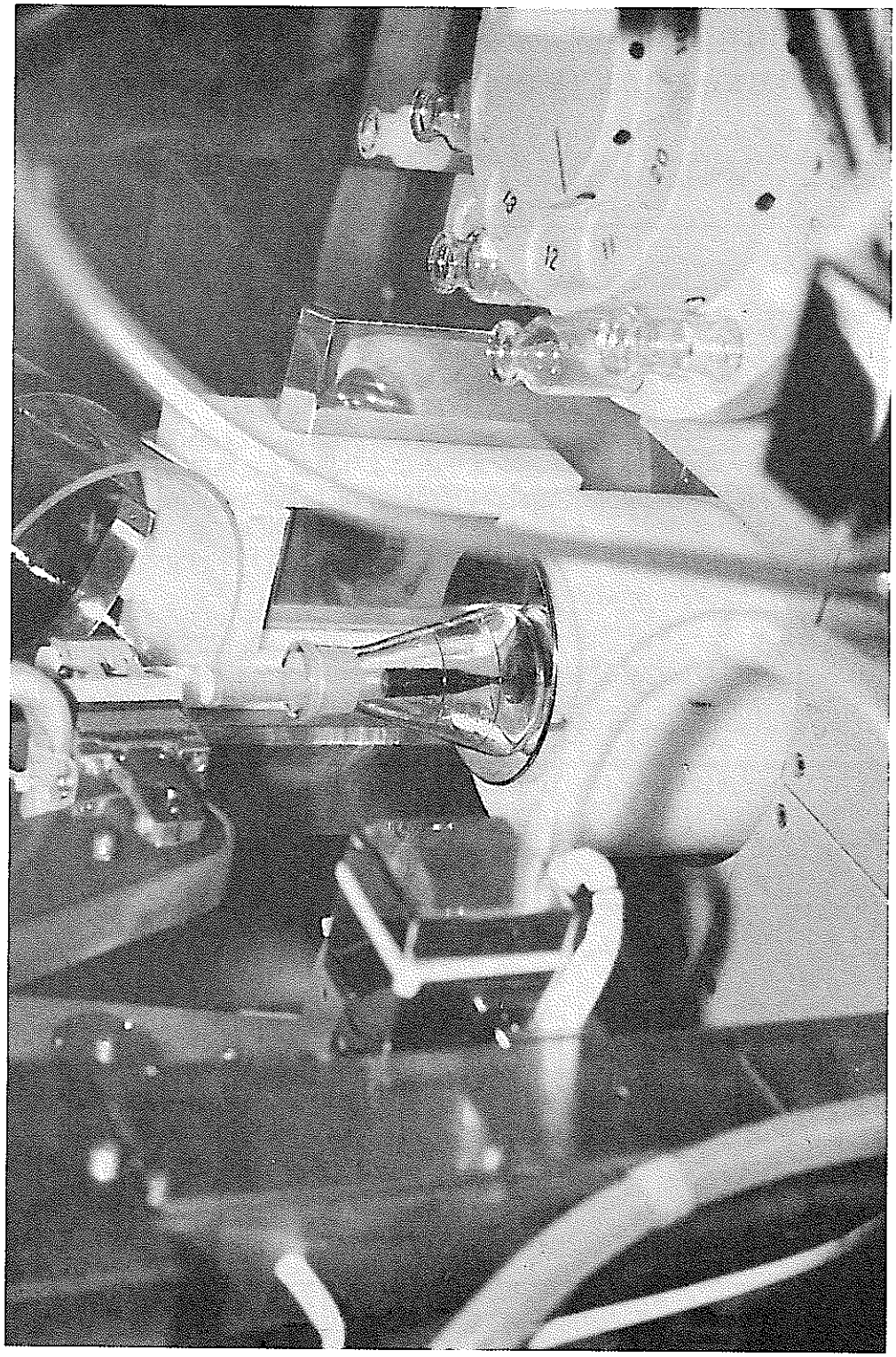


Fig. 27 Plutonium sample to be diluted in a flask

(6) Control unit

The control unit is composed of computer system (HP9816), display panel, floppy disk controller (HP9122D), printer (HP35720A) and interface is shown in Fig. 9 and 28. The unit controls the robot arm, turntables, balance cover, balance and diluter through interfaces.

(7) Software

1) Main program "SAMPLING"

This program operates the first sampling, the dilution, the second sampling and the other essential continuous movements necessary for inspector sample preparation. This software includes pulse setting program by which robot arm position can be described.

2) Movement test program "MOVE TEST"

This program checks the movement of the robot arm, the automated diluter, the turntables. The program list of the main program "SAMPLING" is attached in annex 1 and "MOVE TEST" is in annex 2.

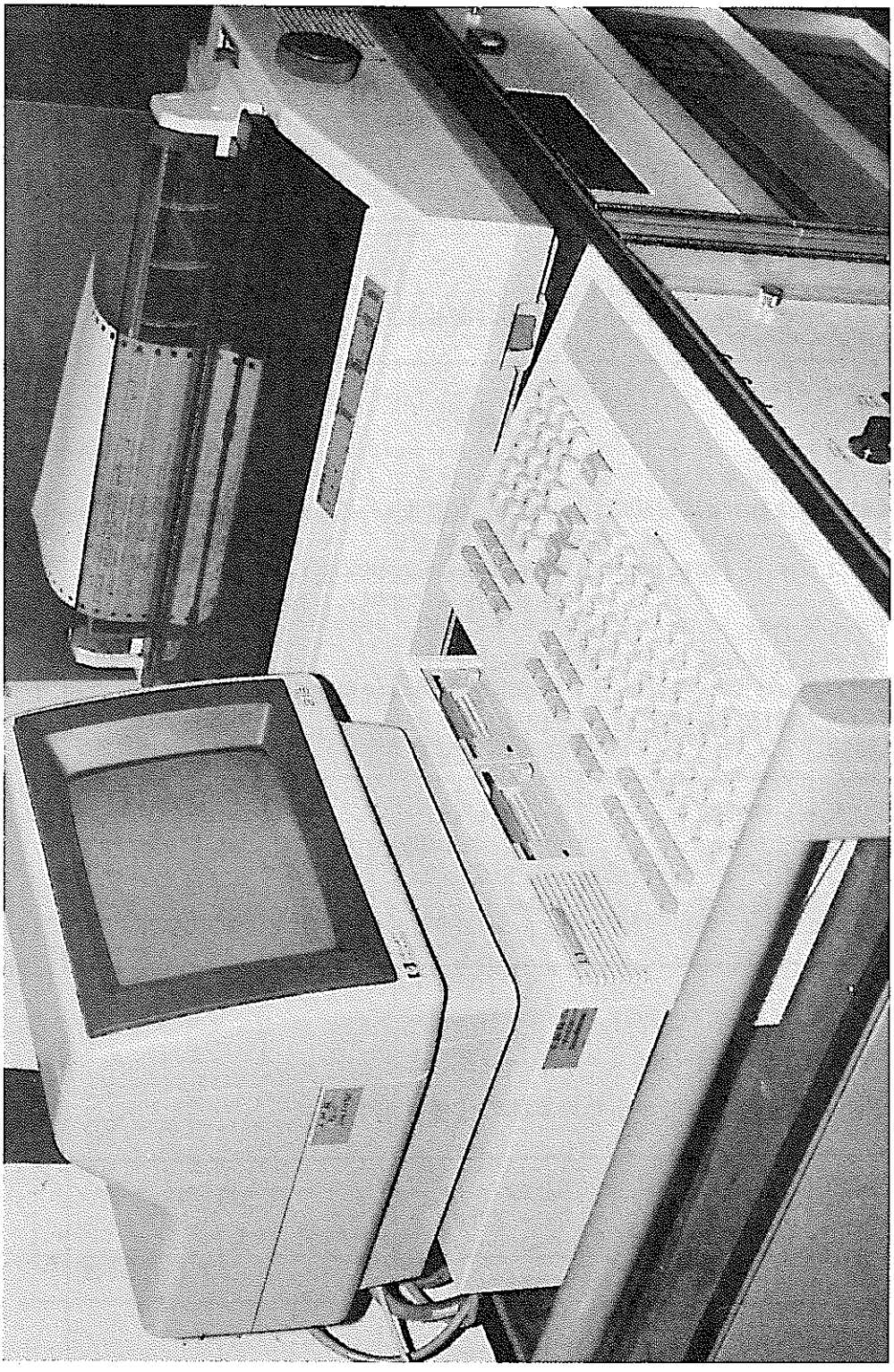


Fig. 28 CPU

4. RESULTS AND DISCUSSIONS

4.1 PERFORMANCE TEST OF THE BALANCE

Because the balance commercially available was separated to two parts, the performance test seemed to be necessary. And also the effect of the magnetic field on the use of balance was checked owing to use of magnetic stirrer.

(1) Repeatability and linearity test

The repeatability of the balance on replicate measurement with 100 gram weight was 0.13mg (1 σ). No difference performance has been detected from the commercial balance (Table 1). The linearity of the balance was also examined. The result is shown in table 2 and correlation coefficient was 0.999999. It was found that the separately controlled balance satisfied the specification neccessary for the gravimetric sample preparation system and gave the same performance as that of the commercial balance.

(2) Influence of the magnetic stirrer on the balance.

Magnetic stirrer is used when the samples are diluted with 3M HNO_3. The influence of the stirrer on the electric balance was examined after setting it either directory on the balance or at the bottom of the flask placed on the balance.

① Influence of the magnetic stirrer set directly on the balance.

The magnetic stirrer, 200mm long, was set on the center of the balance. The influence of the magnetic stirrer was studied.

It was found on the scatter of the replicate neasurements when composed with that examined without stirrers.

② Influence of the magnetic stirrer set in the flask

The magnetic stirrer, the same size as above, and the 10 gram weight was set at the bottom of the flask which was placed on the balance. The measured values with and without the magnetic stirrer (plus flask) were compared. The results is shown in Table 3. No significant difference was detected on both results.

It can be concluded that stirrer put in the flask does not affect the weight measurement by balance.

Table 1. Results of repeatability test for balance

time	Weight, 100g	time	Weight, 100g
1	99.9998	11	99.9996
2	99.9995	12	99.9996
3	99.9998	13	99.9998
4	99.9995	14	99.9996
5	99.9995	15	99.9997
6	99.9994	16	99.9998
7	99.9996	17	99.9997
8	99.9995	18	99.9998
9	99.9996	19	99.9997
10	99.9996	20	99.9998
Av. 99.99965 S.D. 0.00013 CV % 0.0001			

Table 2. Results of linearity test

Total			
weight used	① measured volume	② certified volume	②－① (g) Error
10＋20g	30.0008	30.0008	±0.0000
10＋50g	60.0016	60.0016	±0.0000
10＋100g	110.0024	110.0026	－0.0002
20＋50g	70.0018	70.0018	±0.0000
20＋100g	120.0028	120.0028	±0.0000
50＋100g	150.0037	150.0036	＋0.0001

Correlation coefficient: 0.999999

Table 3. Influence of magnetic stirrer to the balance (set in flask)

Weight	Times	A (g) Weight	B (g) Flask + Stirrer	C (g) Flask + Stirrer + Weight	D=C−B (g) Calculated Weight
10g	1	10.0002	38.8693	48.8697	10.0004
	2	10.0002	38.8690	48.8693	10.0003
	3	10.0002	38.8682	48.8683	10.0001
	4	10.0001	38.8685	48.8686	10.0001
	5	10.0002	38.8682	48.8685	10.0003
	6	10.0002	38.8701	48.8704	10.0003
	7	10.0003	38.8685	48.8688	10.0003
	8	10.0002	38.8704	48.8710	10.0006
	9	10.0002	38.8701	48.8712	10.0002
	10	10.0002	38.8712	48.8715	10.0003
	Ave.	10.0002			10.0003
	S.D.	0.00007			0.0001
	CV %	0.000			0.001

4.2 EVAPORATION OF SOLUTION DURING SAMPLE PERPARATION.

Approximately one hour should be needed for the sample preparation. The rate of evaporation were studid on the diluted solution and each aliquot.

(1) Evaporation of the diluted solution

① The rate of evaporation in the polyethylene cup solution

The rate of evaporation taking place during the second ddilution was investigated. Table 4-a and Fig. 29-A-a give the results of the polythylene cup, widely open of the top of the cup, which shows the rate of evaporation was 0.41% an hour. That seems too large to be neglected for accountability sample preparation.

② The rate of evaporation of the solution in the film coverd cup

The rate of the evaporation of the solution in the film covered cup, which is normally used for the dilution, was investigated (Table 4-b, Fig. 29A-b). The rate of evaporation was approximetly 0.06%/hr that can be regarded relatively small for accountability sample preparation. However, if the tip touches the film at the first sampling, the touch may cause the tip to contaminate at aliquoting form diluted solution.

③ The rate of the evaporation of the solution in a flask (Table 4-c, Fig. 29A-c)

The rate of the evaporation of the solution in a 50 ml flask was 0.02%/hr which can be negligible from the view point of the accountability use. In conclusion, the flask was, therefore, adopted for the dilution.

(2) The rate of the evaporation of the aliquot taken from diluted solution

① The rate of evaporation of the aliquot the vial

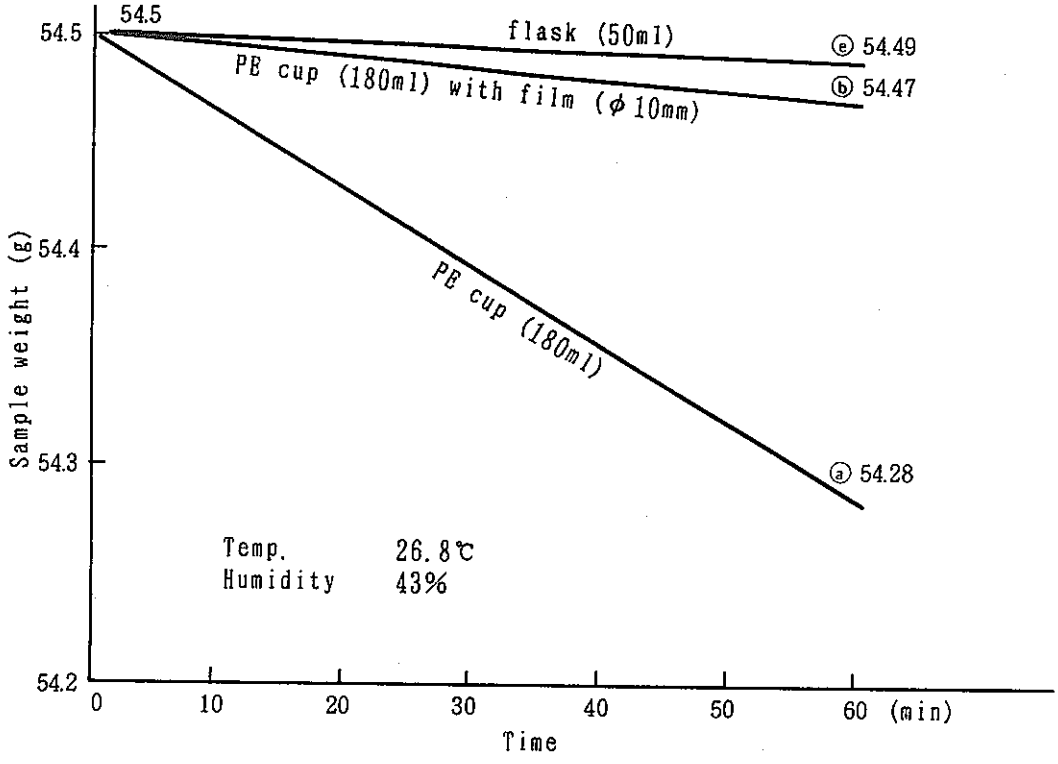
The rate of evaporation of the aliquot in the vial was also studied. It takes 10 to 50 sec. necessary for weighing, which is only time to be considered for evaporation. Therefore the loss based on the evaporation is estimated to be approximately 0.02%. This value is small enough for the accurate sample preparation (Table 4-d, Fig. 29B-d).

② The rate of evaporation of the aliquot in the polyethylene cup

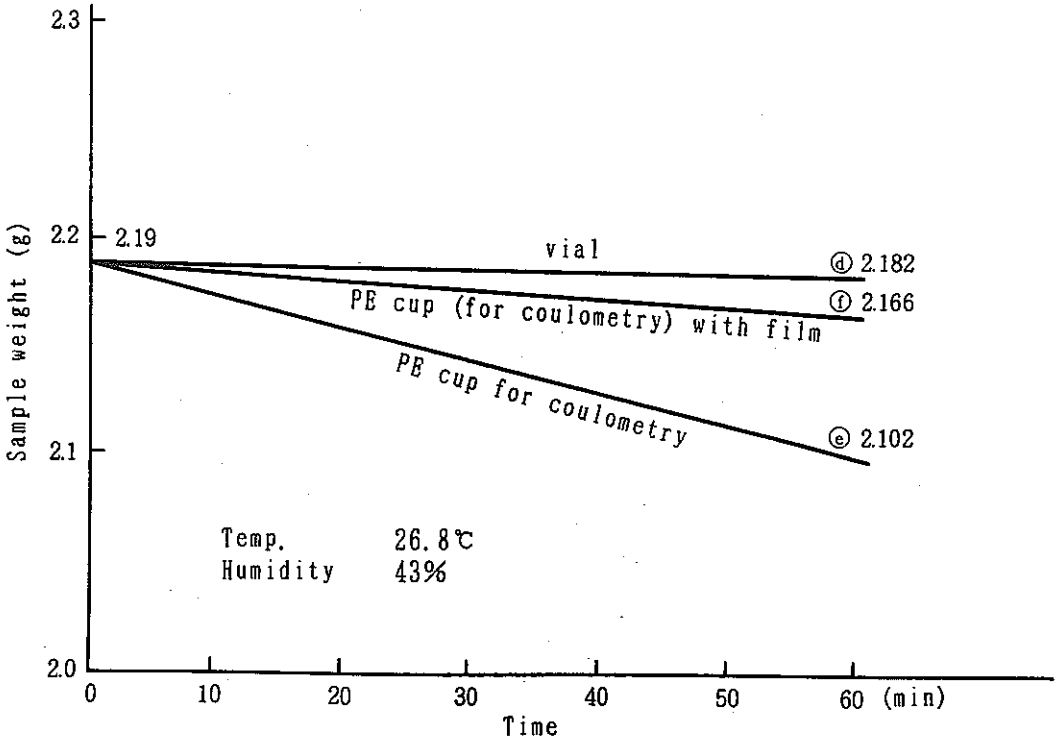
The rate of evaporation of the solution in the polyethylene used for the coulometric determination cup was 0.005%/15 sec. with film cover and 0.02%/15 sec. without it. The rate of the evaporation of it with film cover is small enough for the facility accountability analysis (Table 4-ef, Fig. 29B-ef).

Table 4 Rate of evaporation

	Evaporation rate on dilution cups			Evaporation rate on vials on cup for coulometry		
Time (min)	Cup(a) (120ml)	(Cup)+ (holed film)(b)	Flask(c) (50ml)	Vial (d) (8ml)	Cup for (e) coulo(40ml)	Film covered (f) cup
0	54.50(g)	54.50(g)	54.50(g)	2.190(g)	2.190(g)	2.190
5	54.48	54.50	54.50	2.189	2.183	2.188
10	54.46	54.49	54.50	2.189	2.175	2.186
15	54.45	54.49	54.50	2.188	2.168	2.184
20	54.43	54.49	54.50	2.187	2.160	2.182
25	54.41	54.49	54.50	2.187	2.153	2.180
30	54.39	54.48	54.49	2.186	2.146	2.178
35	54.37	54.48	54.49	2.186	2.139	2.176
40	54.35	54.48	54.49	2.185	2.131	2.174
45	54.34	54.48	54.49	2.184	2.125	2.172
50	54.32	54.47	54.49	2.184	2.117	2.170
55	54.30	54.47	54.49	2.183	2.109	2.168
60	54.28	54.47	54.49	2.182	2.102	2.166
Decreased Weight(g/h)	0.22	0.03	0.01	0.008	0.088	0.024
Decreased ratio (%/h)	0.40	0.06	0.02	0.37	4.0	1.1



A. Rate of evaporation on dilution cups



B. Rate of evaporation on vial/coulo. cup

Fig. 29 Difference of rate of evaporation on various shapes of cells

4.3 THE OPTIMUM CONDITION ON THE MOVEMENT SOFTWARE

The following two sequences were studies to obtained optimum condition on the movement software

(1) Movement sequence (I)

Fig. 30 shows the procedure of the gravimetric movement sequence proposed based on the conventional volumetric method. According to the previous chapter, it can be said that the weight loss on evaporation causes one of the major errors in this system. Therefore any solutions should not be left in open cup. The flow chart of the movement sequence (I) is shown in Fig. 31.

Table 5 is the etimated time consumed during the operation of the programed movement sequence (I). The following errors in sequence (I) were studied.

① The error on the evaporation of the plutonium product

The time to be considered for the evaporation of the concentrated plutonium product solution (t1) does not depend on the number of the dilutions (N) but just the stability time of the balance, i.e. 10 to 15 sec.

It was estimated that the error based on this operation should be 0.0001%.

② The error on the evaporation of the diluted solution

The time to be considered for the evaporation of the diluted solution (t2) depends on the number of diluted solutions (N). The error from the evaporation was estimated 0.027% when three aliquots were taken.

③ The error on the evaporation of the aluquot

The time to be considered for the evaporation of the aliquot taken from the diluted solution (t3) was about 10 to 15 sec., so that the error based on the loss of the solution volume was 0.001%, which seems to be small enough for the accountability.

④ The error on the evaporation observed throughout the total scheme

All movement scheme takes 77, 95, 113 minutes when the number of the dilution as shown in Table 5. The error on the evaporation in the total scheme could not be negligible, if the number of diluted solutions were more than two in the movement sequence (I).

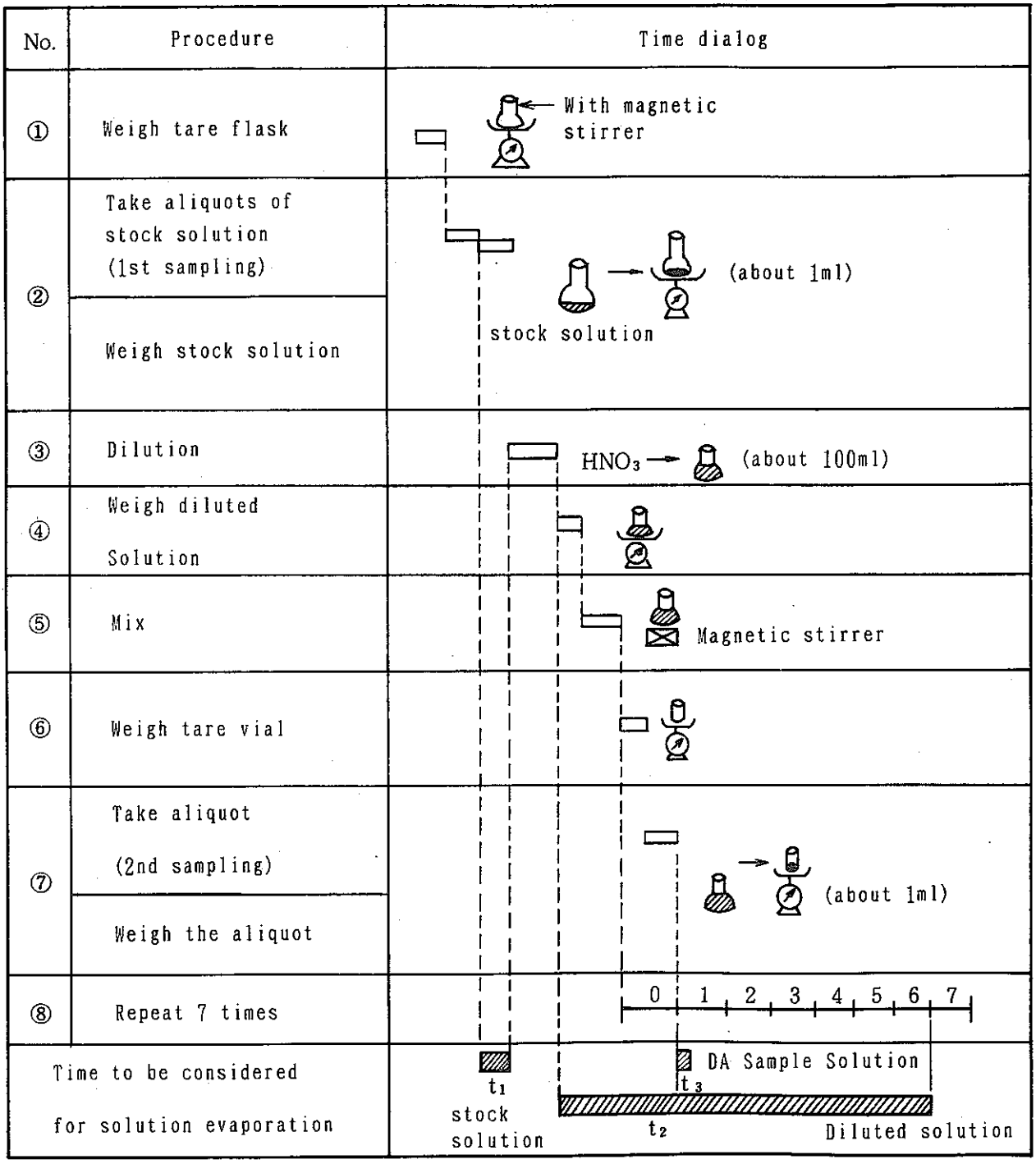


Fig. 30 Procedure of movement sequence

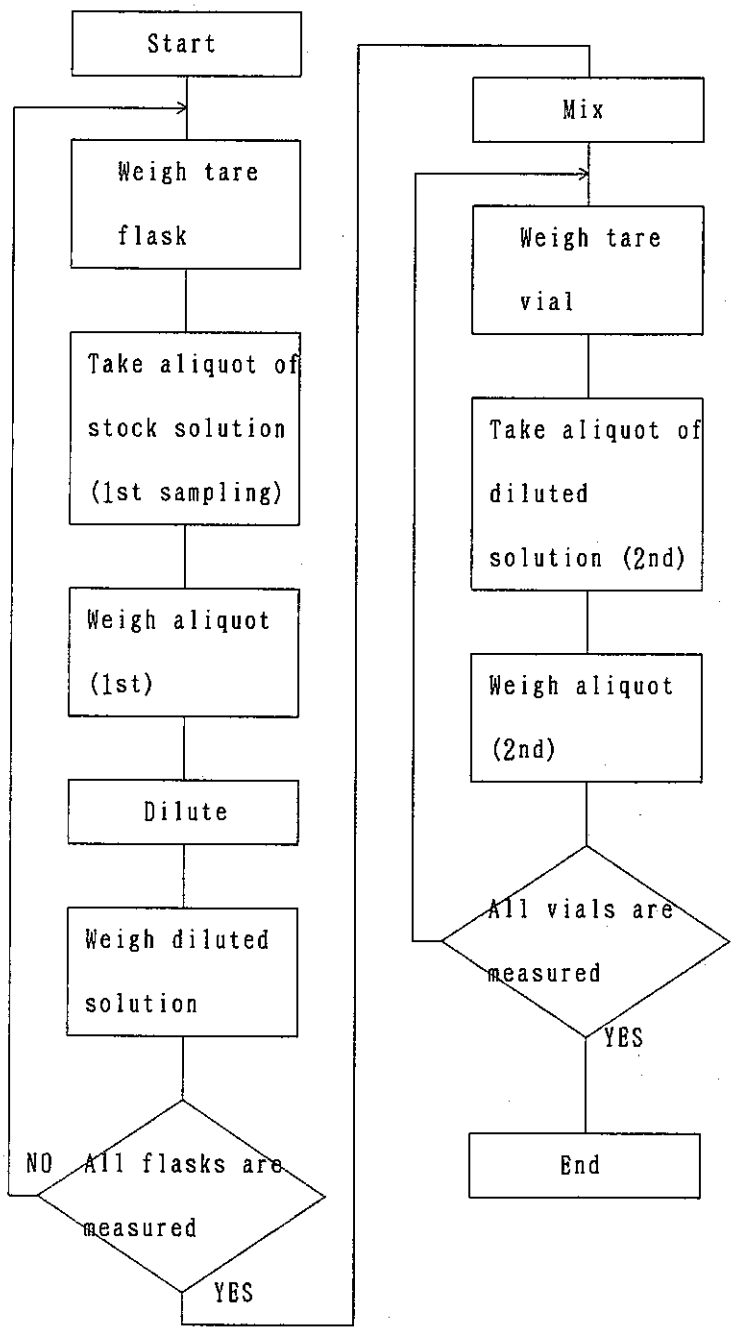


Fig. 31 Movement Sequence (I)

Table 6 Measurement results of Pu Products *

Product No.	Volumetric (Pu g/ℓ)			Gravimetric (Pu g/ℓ)			E%
	n_1	n_2	$\overline{x}$ (R %)	n_1	n_2	$\overline{x}$ (R %)	
1	221.12	221.29	221.21 (0.08)	221.68	221.62	221.65 (0.03)	+0.20
2	206.16	206.23	206.20 (0.03)	206.40	206.42	206.41 (0.01)	+0.10
3	225.36	225.89	225.63 (0.24)	225.69	225.95	225.82 (0.11)	+0.08
4	225.95	225.32	225.13 (0.16)	226.61	226.35	226.48 (0.11)	+0.15
5	223.15	223.44	223.29 (0.13)	223.61	223.81	223.71 (0.09)	+0.19
6	223.74	223.56	223.65 (0.08)	224.18	223.95	224.07 (0.10)	+0.19
7	200.27	200.35	200.31 (0.04)	200.84	200.60	200.72 (0.08)	+0.21
8	205.54	205.67	205.61 (0.06)	206.19	206.01	206.10 (0.09)	+0.24
9	212.87	212.34	212.61 (0.25)	212.52	212.83	212.68 (0.15)	+0.03
10	195.47	195.68	195.58 (0.11)	195.93	196.02	195.98 (0.05)	+0.21
Ave.			$\overline{R}=0.11$ $\sigma_R=0.08$			$\overline{R}=0.08$ $\sigma_R=0.04$	0.16

* The error values in table include those of the coulometic determination

$$E\% = \frac{\overline{x}\ (\text{Gravimetric}) - \overline{x}\ (\text{Volumetric})}{(\overline{x}\ (\text{Gravimetric}) + \overline{x}\ (\text{Volumetric}))\ /\ 2} \times 100$$

(2) Movement sequence (Ⅱ)

In the movement sequence (Ⅱ) shown in Fig. 32. Major difference is the order of the operations; the weight of three independently diluted solution were measured just before taking aliquot to the final vials. The results are shown in Table 5.

The results of the improvement of the sequence were as follows,

① Compared with the movement sequence (Ⅰ), the time to be considered for the operation of the diluted solutions were reduced 12, 61 and 73% at the number of repetition of 1, 2 and 3 respectively.

② With respect to the total time required, it was not extremely shortened.

③ The errors on the evaporation throughout the scheme are 0.019, 0.009, 0.007% at the number of dilution, 1, 2 and 3 respectively.

It can be concluded that the movement sequence (Ⅱ) is applicable to the safeguards and accountability sample preparation.

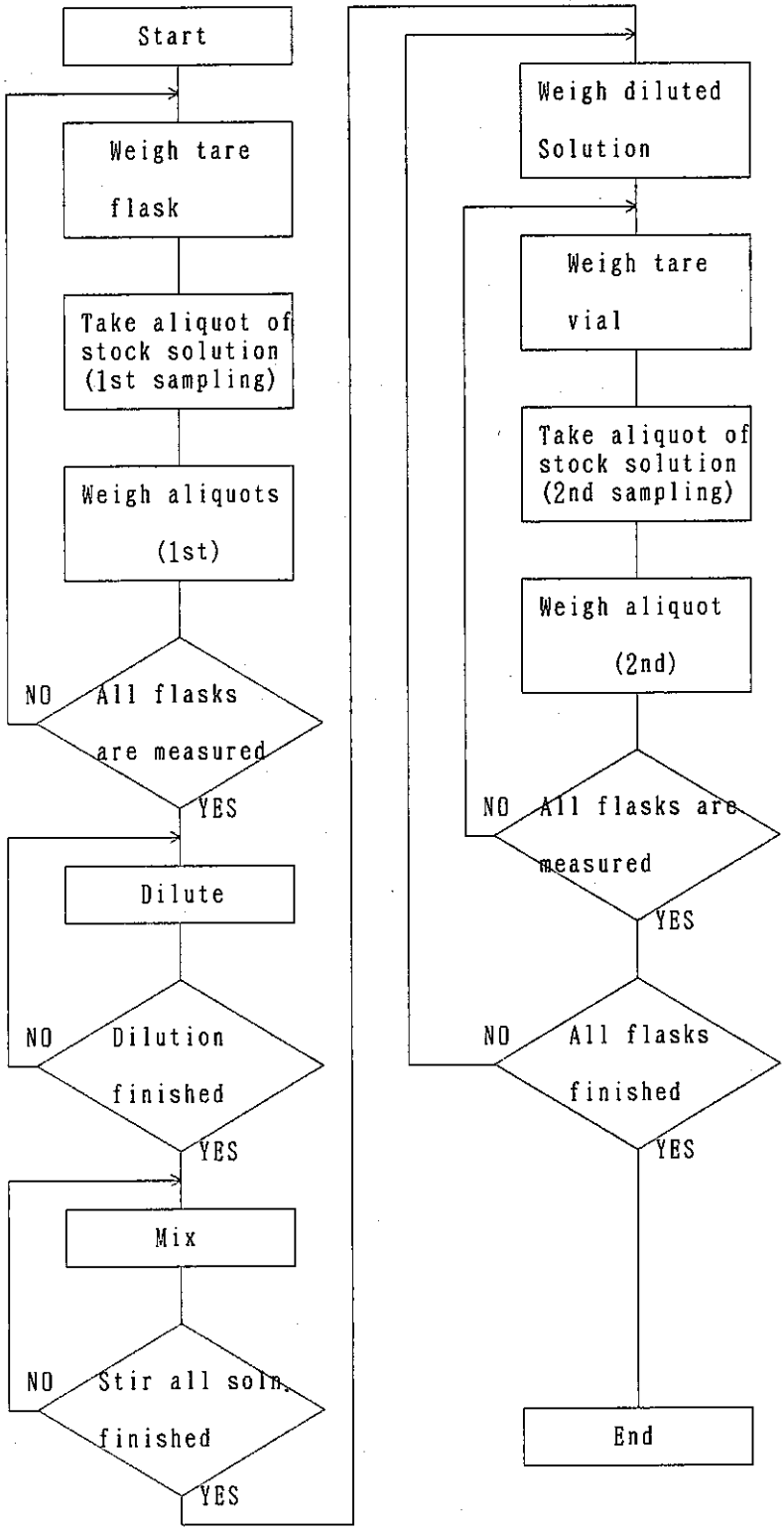


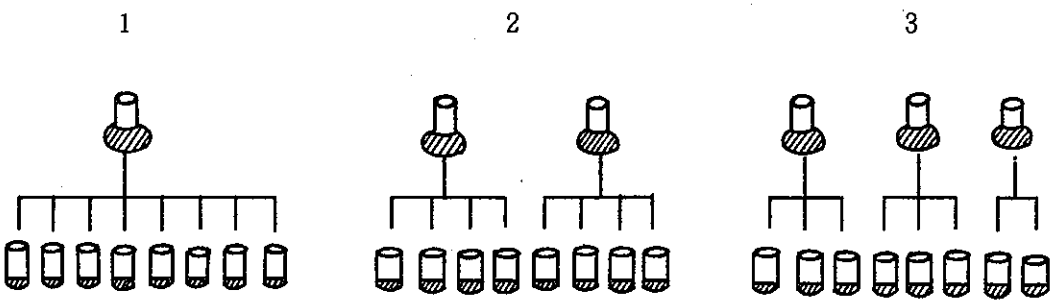
Fig. 32 Movement sequence (Ⅱ)

Table 5 Time to be considered for evaporation of solution on Sequence (Ⅰ) and (Ⅱ)

Mode		Movement sequence (Ⅰ)			Movement sequence (Ⅱ)		
Number of diluted Samples (N)*		1	2	3	1	2	3
Time to be considered (min.)							
1	Taking stock solution (t1)	0.25			0.25		
2	Dilution (t2)	64	72	80	56	28	21
3	Taking aliquots (t3)	0.25			0.25		
Total time (min.)*		77	95	113	79	97	116

* Time required was evapolated in the case 8 vials of aliquots were finally taken. (See figure shown below)

Figure of aliquoting (finally 8 vials)



4.4 APPLICATION TO THE PLUTONIUM PRODUCT SAMPLE

Plutonium concentrations of the product solutions were determined by the coulometry after small aliquots had been prepared by both the volumetric method and the gravimetric one. The errors (%) estimated from duplicate results were 0.08% and 0.04% on the analytical data whose sample preparations were performed by volumetric and gravimetric method, respectively, whereas about 0.2% bias was observed between the values of the volumetric and the gravimetric method. In addition the value based on the volumetric method always tends to be lower than that on gravimetric one. This seems to be mainly due to the fact the pipets used for the volumetric sample preparation were calibrated with water. In the other words, the amount of remaining concentrated plutonium attached to the inside of the pipets may be different from that on the calibration with water, which can cause bias.

In the case of the gravimetric mathod was used, plutonium amount is expressed as follows;

$$Pu(g) = V(\ell) \times \rho\,(g\ soln./\ \ell) \times Pu(g/g\ soln.)$$

The density must be measured at 31 ℃ in the laboratory, because the manometer used for the volume measurement is calibrated by useig at 31℃, and the volume data obtained routinly are normalized to those at 31 ℃. Therefore there is no factor to generate a bias based on the difference of temperature, which is a major advantage on the gravimetric method.

5. CONCLUSION

(1) The Automated Gravimetric Sample preparation System (AGSS) for the plutonium product solutions has been developed. This method needs 70 minutes to prepare samples.

(2) The error estimated based on the preparation scheme is around 0.02% that is due to the error based on the solutions evaporation observed throughout the sample preparation.

(3) Compared with the measured plutonium value obtained based on the gravimetric method, 0.16% bias in average of 10 samples. It can be thought that this error was due to the remaining plutonium amount attached to the inside of the pipet on the volumetric method.

It is concluded that the automated gravimetric sample preparation system for the plutonium product solution can provides accurate and reliable sample preparation independently of the temperature effect.

6. FUTURE PLAN

The automated gravimetric sample preparation system for the plutonium product solutions has been implemented used at the Tokai Reprocessing Plant for the accountability and safeguards purpose from the PIV December 1988. The automated gravimetric sample preparation system for input solutions has been also developed. The system had been completed and cold tests have been carried out. The application to the use of the LSD, large size dried spike, however, should be also studied because of the modification of the sample preparation scheme. This system will be installed in the shielding cell in 1990.

REFERENCE

1) "The JASPAS", Compiled by Nuclear Material Control Center, March 1988.

2) PNC SN8520 86-19 "The Control Manual for Accountancy Analysis", March 1986.

3) PNC SN841 85-02 "Controlled Potential Coulometry for Plutonium Product of Reprocessing Plant", January 1985.

4) PNC SN8520 88-003 "Handling Manual of the Automated Gravimetric Sampling System for the Plutonium Product Accountancy Sample Preparation", July 1988.

5) Y. Kuno, K. Abe et al. "Use of Automated Controlled Potential Coulometry System for Plutonium Product Accountancy Analysis at Tokai Reprocessing Plant", P859 29th INMM ANNUAL MEETING PROCEEDING, June 1988.

List of main program "SAMPLING"

```
1000  !*****************************************************************
1005  !*                                                               *
1010  !*   Automated Gravimetric Sampling System                       *
1015  !*                                                               *
1020  !*                                  Rev.0  1985-12-05  by NCS    *
1025  !*                                  Rev.1 *1986-10-08            *
1030  !*                                  Rev.2 *1988-02-18            *
1035  !*                                  Rev.3 *1988-09-02  Vial only*
1040  !*                                  Rev.4 *1988-12-10            *
1045  !*                                                  by Abe    *
1050  !*****************************************************************
1055  !
1060  OPTION BASE 1
1065  COM /Chit_data/ Unit$(2)[10],Chit$(2)[8],Name$(2)[10],Date$(2)[8]
1070  COM /Sample_1/ REAL S1_sample(2,4),S1_dilute(2,4)
1075  COM /Sample_2/ INTEGER S2_cup(2,20),S2_numbers(2,20)
1080  COM /Sample_2/ REAL S2_sample(2,20)
1085  COM /Bottle_data/ Bottle$(20)[6],REAL Inp_vial(20)
1090  COM /Index_data/ INTEGER File_num,Idx_chit$(195)[8],Idx_unit$(195)[10]
1095  COM /Index_data/ Idx_name$(195)[10],Idx_date$(195)[8]
1100  COM /File_data/ Chit_x$[8],Unit_x$[10],Name_x$[10],Date_x$[8]
1105  COM /File_data/ REAL Sample_1(4),Dilute(4)
1110  COM /File_data/ REAL Sample_2(20),INTEGER Cup_num(20),Bottles$(20)[6]
1115  COM /File_data/ REAL Inp_vials(20),Mea_vials(20)
1120  COM /Get_data/ INTEGER Max_page,Max_data
1125  COM /Get_data/ REAL Sample_1x,Dilute_x,Dilute_rate
1130  COM /Get_data/ REAL Sample_2x(20),Bottle_x$(20)[6]
1135  COM /Gmax_io/ @Gmax_io
1140  COM /Gmax_pm/ @Gmax_pm
1145  COM /Rs_232c/ @Rs_232c
1150  COM /Result/ REAL Sample_1st(2,4),Dilution(2,4)
1155  COM /Result/ REAL Sample_2nd(2,20),INTEGER Cup_no(2,20),REAL Mea_vial(20)
1160  COM /Measure/ INTEGER Solution,Sampling,Max_cup,Max_bottle
1165  COM /Measure/ INTEGER Now_cup,Now_bottle,S2_point
1170  COM /T_table/ INTEGER C_posi,B_posi
1175  COM /Robot/ INTEGER Now_posi,Hand,REAL Length,Hight,Turn
1180  COM /Message/ INTEGER Msg_flag
1185  !
1190  INTEGER I,J,Key_no,Ret_code
1195  !
1200  !---------- start program ------------------------
1205  !
1210 Start:!
1215  PRINTER IS 30
1220  PRINT CHR$(27)&"B"&CHR$(72);
1225  PRINTER IS CRT
1230  GRAPHICS ON
1235  OUTPUT 2;"SCRATCH KEYX";
1240 !CALL K_init
1245 !CALL K_font_is("DOT","9816S")
1250 !CALL K_csize(.001)
1255 !CALL K_space_ratio(.001)
1260 !CSIZE 5,.6
1265  GCLEAR
1270  VIEWPORT 0,399,0,299
1275  WINDOW 0,399,0,299
1280  !
1285  Msg_flag=0
1290  CALL Get_dtfile(Ret_code)
1295  IF Ret_code<>0 THEN
1300     CALL Msg_disp(Ret_code)
1305     STOP
1310  END IF
1315  CALL Initial(Ret_code)
1320  IF Ret_code<>0 THEN
1325     CALL Msg_disp(Ret_code)
1330     STOP
1335  END IF
```

```
1340  CALL Set_date(Ret_code)
1345  !
1350  GOTO 1385
1355  !GOTO Test2
1360 Test1:!
1365  CALL Test10
1370 Test2:!
1375  CALL Test20
1380  !
1385  LOOP
1390     ON KNOB 1.0 CALL Dummy
1395     PRINTER IS 1
1400     CALL Disp_menu(1)
1405     IF Ret_code<>0 THEN
1410        CALL Msg_disp(Ret_code)
1415     END IF
1420     Inp$=""
1425     CALL Key_in(1,Inp$,Inp$,Key_no)
1430     SELECT Key_no
1435     CASE =0
1440        CALL Set_data(Ret_code)
1445     CASE =1
1450        CALL Measure(Ret_code)
1455     CASE =2
1460        CALL Result_data(Ret_code)
1465     END SELECT
1470  END LOOP
1475  END
1480  !
1485  SUB Initial(INTEGER Ret_code)
1490  !*****************************************************
1495  !*
1500  !*       Initialize Device
1505  !*
1510  !*****************************************************
1515     OPTION BASE 1
1520     COM /Gmax_io/ @Gmax_io
1525     COM /Gmax_pm/ @Gmax_pm
1530     COM /Rs_232c/ @Rs_232c
1535     !
1540 Initial:!
1545     Gmax_io=709
1550     Gmax_pm=710
1555     Rs_232c=9
1560     CONTROL Rs_232c,3;2400
1565     CONTROL Rs_232c,4;2+0+8+16
1570     ASSIGN @Rs_232c TO Rs_232c
1575     ASSIGN @Gmax_io TO Gmax_io
1580     ASSIGN @Gmax_pm TO Gmax_pm
1585     !
1590     ON TIMEOUT SC(@Gmax_io),10.0 GOTO Timeout
1595     S=SPOLL(@Gmax_io)
1600     CLEAR @Gmax_io
1605     WAIT 5.0
1610     OUTPUT @Gmax_io;"ST,"&VAL$(IVAL("01101100",2))
1615     OFF TIMEOUT
1620     !
1625     ON TIMEOUT SC(@Gmax_pm),10.0 GOTO Timeout
1630     S=SPOLL(@Gmax_pm)
1635     CLEAR @Gmax_pm
1640     WAIT 5.0
1645     OUTPUT @Gmax_pm;"ST,"&VAL$(IVAL("01101100",2))
1650     OFF TIMEOUT
1655     !
1660     CALL Tenbin(0,0,Ret_code)
1665     IF Ret_code<>0 THEN GOTO Timeout
1670     CALL Titrater(0,0,Ret_code)
1675     IF Ret_code<>0 THEN GOTO Timeout
1680     CALL Motor_1(0,0,Ret_code)
1685     IF Ret_code<>0 THEN GOTO Timeout
1690     CALL Motor_2(0,0,Ret_code)
1695     IF Ret_code<>0 THEN GOTO Timeout
```

```
1700      CALL Motor_3(0,0,Ret_code)
1705      IF Ret_code<>0 THEN GOTO Timeout
1710      CALL Motor_4(0,0,Ret_code)
1715      IF Ret_code<>0 THEN GOTO Timeout
1720      CALL Motor_5(0,0,Ret_code)
1725      IF Ret_code<>0 THEN GOTO Timeout
1730      CALL Move_hand(0,0,Ret_code)
1735      IF Ret_code<>0 THEN GOTO Timeout
1740      !
1745      CALL Status(0,Ret_code)
1750      SUBEXIT
1755  Timeout:!
1760      OFF TIMEOUT
1765      Ret_code=8
1770   SUBEND
1775   !
1780   SUB Set_data(INTEGER Ret_code)
1785   !*********************************************************
1790   !*
1795   !*    Set Measurement-parameter
1800   !*
1805   !*********************************************************
1810      OPTION BASE 1
1815      COM /Chit_data/ Unit$(*),Chit$(*),Name$(*),Date$(*)
1820      INTEGER Key_no,Ret
1825     !
1830      Ret=0
1835      Key_no=0
1840      WHILE Key_no<>9
1845         CALL Disp_menu(2)
1850         IF Ret<>0 THEN
1855            CALL Msg_disp(Ret)
1860         END IF
1865         Inp$=""
1870         CALL Key_in(2,Inp$,Inp$,Key_no)
1875         SELECT Key_no
1880         CASE =0
1885            CALL Set_sample(1)
1890         CASE =1
1895            CALL Set_sample(2)
1900         CASE =2
1905            CALL Set_bottle(1,Key_no)
1910            Key_no=2
1915         CASE =3
1920            IF Chit$(1)<>RPT$(" ",8) THEN
1925               CALL Print_sample(1,Ret)
1930            END IF
1935            IF Chit$(2)<>RPT$(" ",8) THEN
1940               CALL Print_sample(2,Ret)
1945            END IF
1950            CALL Print_bottle(Ret)
1955         END SELECT
1960      END WHILE
1965      Ret_code=0
1970   SUBEND
1975   !
1980   SUB Measure(INTEGER Ret_code)
1985   !*********************************************************
1990   !*
1995   !*       Sample Measurement
2000   !*
2005   !*********************************************************
2010      OPTION BASE 1
2015      COM /Chit_data/ Unit$(*),Chit$(*),Name$(*),Date$(*)
2020      INTEGER Key_no
2025     !
2030      Key_no=0
2035      WHILE Key_no<>9
2040         CALL Disp_menu(3)
2045         IF Ret_code<>0 THEN
2050            CALL Msg_disp(Ret_code)
2055         END IF
```

```
2060        Inp$=""
2065        CALL Key_in(6,Inp$,Inp$,Key_no)
2070        SELECT Key_no
2075        CASE =0
2080           CALL Check_data(Ret_code)
2085           IF Ret_code=0 THEN
2090              CALL Put_dtfile(Ret_code)
2095              IF Ret_code=0 THEN
2100                 CALL State_samp(Ret_code)
2105                 IF Ret_code=0 THEN
2110                    CALL Put_file(1,Ret_code)
2115                    IF Ret_code=0 THEN
2120                       CALL Put_file(2,Ret_code)
2125                       IF Ret_code=0 THEN
2130                          !
2135                          IF Chit$(1)<>RPT$(" ",8) THEN
2140                             CALL Get_index(Ret_code)
2145                             IF Ret_code=0 THEN
2150                                CALL Get_file(Chit$(1),Ret_code)
2155                                IF Ret_code=0 THEN
2160                                   CALL Print_result(Ret_code)
2165                                END IF
2170                             END IF
2175                          END IF
2180                          !
2185                          IF Chit$(2)<>RPT$(" ",8) THEN
2190                             CALL Get_index(Ret_code)
2195                             IF Ret_code=0 THEN
2200                                CALL Get_file(Chit$(2),Ret_code)
2205                                IF Ret_code=0 THEN
2210                                   CALL Print_result(Ret_code)
2215                                END IF
2220                             END IF
2225                          END IF
2230                          !
2235                       END IF
2240                    END IF
2245                 END IF
2250              END IF
2255           END IF
2260        CASE =1
2265           CALL Msg_disp(15)
2270           CALL Titrater(1,20,Ret_code)
2275           CALL Msg_disp(0)
2280        END SELECT
2285     END WHILE
2290     Ret_code=0
2295  SUBEND
2300  !
2305  SUB Result_data(INTEGER Ret_code)
2310  !*************************************************************
2315  !*
2320  !*   Display or Print the Result-Data
2325  !*
2330  !*************************************************************
2335     OPTION BASE 1
2340     COM /Chit_data/ Unit$(*),Chit$(*),Name$(*),Date$(*)
2345     COM /File_data/ Chit_x$,Unit_x$,Name_x$,Date_x$
2350     COM /File_data/ REAL Sample_1(*),Dilute(*)
2355     COM /File_data/ REAL Sample_2(*),INTEGER Cup_num(*),Bottles$(*)
2360     COM /File_data/ REAL Inp_vials(*),Mea_vials(*)
2365     DIM Guide$[50]
2370     INTEGER Key_no,Cup,Page,Max_page,Ret
2375    !
2380     Key_no=99
2385     WHILE Key_no<>9
2390        IF Key_no=99 THEN
2395           CALL Disp_menu(4)
2400           IF Ret<>0 THEN
2405              CALL Msg_disp(Ret)
2410              Ret=0
2415           END IF
```

```
2420            Inp$=""
2425            CALL Key_in(7,Inp$,Inp$,Key_no)
2430         END IF
2435         SELECT Key_no
2440         CASE 0
2445            CALL Get_index(Ret)
2450            IF Ret=0 THEN
2455               Page=1
2460               WHILE Key_no<>9
2465                  CALL Disp_index(Page)
2470                  Inp$=""
2475                  CALL Key_in(8,Inp$,Inp$,Key_no)
2480                  IF Key_no=0 THEN
2485                     Page=Page+1
2490                  END IF
2495               END WHILE
2500            END IF
2505            Key_no=99
2510         CASE 1
2515            IF Chit$(1)=RPT$(" ",8) THEN
2520               Inp$=Chit$(2)
2525            ELSE
2530               Inp$=Chit$(1)
2535            END IF
2540            Guide$="<Display> File name (Report no.) : "
2545            CALL Key_in(7,Inp$,Guide$,Key_no)
2550            IF Key_no<0 THEN
2555               WHILE Inp$[1;1]=" " AND Inp$<>RPT$(" ",8)
2560                  Inp$=Inp$[2;7]&" "
2565               END WHILE
2570               CALL Get_index(Ret)
2575               IF Ret=0 THEN
2580                  CALL Get_file(Inp$,Ret)
2585                  IF Ret=0 THEN
2590                     Page=1
2595                     WHILE Key_no<>9
2600                        CALL Disp_result(Page)
2605                        Inp$=""
2610                        CALL Key_in(8,Inp$,Inp$,Key_no)
2615                        IF Key_no=0 THEN
2620                           Page=Page+1
2625                        END IF
2630                     END WHILE
2635                  END IF
2640               END IF
2645               Key_no=99
2650            END IF
2655         CASE 2
2660            IF Chit$(1)="          " THEN
2665               Inp$=Chit$(2)
2670            ELSE
2675               Inp$=Chit$(1)
2680            END IF
2685            Guide$="<Print> File name (Report no.) : "
2690            CALL Key_in(6,Inp$,Guide$,Key_no)
2695            IF Key_no<0 THEN
2700               WHILE Inp$[1;1]=" " AND Inp$<>RPT$(" ",8)
2705                  Inp$=Inp$[2;7]&" "
2710               END WHILE
2715               CALL Get_index(Ret)
2720               IF Ret=0 THEN
2725                  CALL Get_file(Inp$,Ret)
2730                  IF Ret=0 THEN
2735                     CALL Print_result(Ret)
2740                  END IF
2745               END IF
2750               Key_no=99
2755            END IF
2760         CASE 3
2765            CALL Make_disk(Ret)
2770            Key_no=99
2775         END SELECT
```

```
2780     END WHILE
2785     Ret_code=0
2790  SUBEND
2795  !
2800  SUB Disp_menu(Menu_no)
2805  !*********************************************************
2810  !*
2815  !*       Display  Menu
2820  !*
2825  !*********************************************************
2830     DIM Buf$[50]
2835     INTEGER I
2840     !
2845 Disp_menu:!
2850     GCLEAR
2855     OUTPUT 2;"K";
2860     MOVE 45,299
2865     RPLOT 0,0
2870     RPLOT 310,0
2875     RPLOT 310,-58
2880     RPLOT 0,-58
2885     RPLOT 0,0
2890     SELECT Menu_no
2895     CASE =1
2900        RESTORE Menu1
2905        READ Buf$
2910       !CSIZE 6,.4
2915       !MOVE 70,264
2920       !LABEL Buf$
2925        PRINT TABXY(20,3);Buf$
2930        FOR I=1 TO 3
2935          !CSIZE 5,.6
2940          !MOVE 60,224-24*I
2945          !LABEL "k"&VAL$(I-1)
2950           READ Buf$
2955          !CSIZE 6,.4
2960          !MOVE 90,224-24*I
2965          !LABEL Buf$
2970           PRINT TABXY(20,2*I+6);"k"&VAL$(I-1);"    ";Buf$
2975        NEXT I
2980     CASE =2
2985        RESTORE Menu2
2990        READ Buf$
2995       !CSIZE 6,.4
3000       !MOVE 110,264
3005       !LABEL Buf$
3010        PRINT TABXY(28,3);Buf$
3015        FOR I=1 TO 4
3020          !CSIZE 5,.6
3025          !MOVE 70,224-24*I
3030          !LABEL "k"&VAL$(I-1)
3035           READ Buf$
3040          !CSIZE 6,.4
3045          !MOVE 110,224-24*I
3050          !LABEL Buf$
3055           PRINT TABXY(23,2*I+6);"k"&VAL$(I-1);"    ";Buf$
3060        NEXT I
3065     CASE =3
3070        RESTORE Menu3
3075        READ Buf$
3080       !CSIZE 6,.4
3085       !MOVE 160,264
3090       !LABEL Buf$
3095        PRINT TABXY(36,3);Buf$
3100     CASE =4
3105        RESTORE Menu4
3110        READ Buf$
3115       !CSIZE 6,.4
3120       !MOVE 60,264
3125       !LABEL Buf$
3130        PRINT TABXY(22,3);Buf$
3135        FOR I=1 TO 4
```

```
3140          !CSIZE 5,.6
3145          !MOVE 70,224-24*I
3150          !LABEL "k"&VAL$(I-1)
3155           READ Buf$
3160          !CSIZE 6,.4
3165          !MOVE 110,224-24*I
3170          !LABEL Buf$
3175           PRINT TABXY(24,2*I+6);"k"&VAL$(I-1);"    ";Buf$
3180        NEXT I
3185     END SELECT
3190     SUBEXIT
3195  !
3200  !------ Data string -----
3205  !
3210 Menu1:!
3215     DATA "Automated Gravimetric Sampling ;  MENU"
3220     DATA "Preset Measuring Data"
3225     DATA "Measure"
3230     DATA "Display and Print Out Measurement Result"
3235 Menu2:!
3240     DATA "Preset Measurement Data"
3245     DATA "Preset Data of Dilution A"
3250     DATA "Preset Data of Dilution B"
3255     DATA "Preset Number of Vials"
3260     DATA "Print Out Preset Data"
3265 Menu3:!
3270     DATA "Measure"
3275 Menu4:!
3280     DATA "Display and Print Out Measurement Result"
3285     DATA "Display Files"
3290   ! DATA "Display Measurement Result"
3295     DATA "                          "
3300     DATA "Print Out Measurement Result"
3305     DATA "Format Disk for Data"
3310  SUBEND
3315  !
3320  SUB Set_sample(INTEGER Solution)
3325  !*********************************************
3330  !*
3335  !*     Set the data for sampling
3340  !*
3345  !*********************************************
3350     OPTION BASE 1
3355     COM /Chit_data/ Unit$(*),Chit$(*),Name$(*),Date$(*)
3360     COM /Sample_1/ REAL S1_sample(*),S1_dilute(*)
3365     COM /Sample_2/ INTEGER S2_cup(*),S2_numbers(*),REAL S2_sample(*)
3370     DIM Guide$[30],Inp$[20]
3375     INTEGER X,Y,No,Key_no
3380     INTEGER Temp1,Temp2
3385     REAL Temp3,Temp4,Temp5
3390     !
3395 Set_sample:!
3400     OUTPUT 2;"K";
3405     GCLEAR
3410     IF Solution=1 THEN
3415       !Guide$="861U!J#A!K7WB,M@%G!<%?@_Dj"
3420        Guide$="Preset Data of Dilution A "
3425     ELSE
3430       !Guide$="861U!J#B!K7WB,M@%G!<%?@_Dj"
3435        Guide$="Preset Data of Dilution B "
3440     END IF
3445    !CALL K_move(28,92)
3450    !CALL K_label(Guide$)
3455     PRINT TABXY(25,1);Guide$
3460     MOVE 7,246
3465     GOSUB Frame_1
3470     PRINT TABXY(3,6);" Unit No.  ";TABXY(17,6);Unit$(Solution)
3475     PRINT TABXY(3,7);" Report No.";TABXY(17,7);Chit$(Solution)
3480     PRINT TABXY(3,8);" Operator ";TABXY(17,8);Name$(Solution)
3485     MOVE 12,174
3490     GOSUB Frame_2
3495     PRINT TABXY(10,10);"1st Sampling"
```

```
3500     PRINT TABXY(4,12);"cup  sampling  dilution"
3505     PRINT TABXY(4,13);" No.   (ml)      (ml)  "
3510     FOR No=1 TO 4
3515        PRINT TABXY(6,No+14);VAL$(No)
3520        Temp4=S1_sample(Solution,No)
3525        Temp5=S1_dilute(Solution,No)
3530        IF Temp4>=0 THEN
3535           PRINT TABXY(12,No+14);RPT$(" ",4-LEN(VAL$(Temp4)))&VAL$(Temp4)
3540           PRINT TABXY(20,No+14);RPT$(" ",5-LEN(VAL$(Temp5)))&VAL$(Temp5)
3545        END IF
3550     NEXT No
3555     MOVE 162,246
3560     GOSUB Frame_3
3565     MOVE 287,246
3570     GOSUB Frame_3
3575     PRINT TABXY(50,4);"2nd Sampling"
3580     PRINT TABXY(34,6);"cup  sampling Number     cup  sampling Number "
3585     PRINT TABXY(34,7);" No.   (ml)   of Vial     No.   (ml)   of Vial"
3590     FOR No=1 TO 20
3595        Temp1=S2_cup(Solution,No)
3600        Temp2=S2_numbers(Solution,No)
3605        Temp3=S2_sample(Solution,No)
3610        IF Temp1<>0 THEN
3615           X=25*((No-1) DIV 10)
3620           Y=((No-1) MOD 10)+9
3625           PRINT TABXY(X+36,Y);VAL$(Temp1)
3630           PRINT TABXY(X+41,Y);RPT$(" ",4-LEN(VAL$(Temp3)))&VAL$(Temp3)
3635           PRINT TABXY(X+51,Y);RPT$(" ",2-LEN(VAL$(Temp2)))&VAL$(Temp2)
3640        END IF
3645     NEXT No
3650     !
3655     Key_no=0
3660     WHILE Key_no<>9
3665        SELECT Key_no
3670        CASE =0
3675           Key_no=-1
3680           No=1
3685           WHILE Key_no<0 OR (Key_no>3 AND Key_no<9)
3690              SELECT No
3695              CASE =1
3700                 Guide$="Unit No. : "
3705                 Inp$=Unit$(Solution)
3710              CASE =2
3715                 Guide$="Report No. : "
3720                 Inp$=Chit$(Solution)
3725              CASE =3
3730                 Guide$="Name : "
3735                 Inp$=Name$(Solution)
3740              END SELECT
3745              PRINT TABXY(1,No+5);"*";
3750              CALL Key_in(3,Inp$,Guide$,Key_no)
3755              PRINT TABXY(1,No+5);" ";
3760              IF Key_no<0 THEN
3765                 SELECT No
3770                 CASE =1
3775                    Unit$(Solution)=Inp$
3780                 CASE =2
3785                    WHILE Inp$[1;1]=" " AND Inp$<>RPT$(" ",8)
3790                       Inp$=Inp$[2;7]&" "
3795                    END WHILE
3800                    Chit$(Solution)=Inp$
3805                 CASE =3
3810                    Name$(Solution)=Inp$
3815                 END SELECT
3820                 PRINT TABXY(17,No+5);Inp$
3825                 No=(No MOD 3)+1
3830              END IF
3835           END WHILE
3840        CASE =1
3845           Key_no=-1
3850           No=1
3855           WHILE Key_no<0 OR (Key_no>3 AND Key_no<9)
```

```
3860 S1_input:!
3865                ON ERROR GOTO Valid_data1
3870                Guide$="(Cup #"&VAL$(No)&")Sampling,Dilution : "
3875                Temp4=S1_sample(Solution,No)
3880                Temp5=S1_dilute(Solution,No)
3885                Inp$=RPT$(" ",4-LEN(VAL$(Temp4)))&VAL$(Temp4)&","
3890                Inp$=Inp$&RPT$(" ",5-LEN(VAL$(Temp5)))&VAL$(Temp5)
3895                PRINT TABXY(2,No+14);"*";
3900                CALL Key_in(3,Inp$,Guide$,Key_no)
3905                PRINT TABXY(2,No+14);" ";
3910                IF Key_no<0 THEN
3915                   IF POS(Inp$,",")=5 THEN
3920                      Temp4=VAL(Inp$[1;4])
3925                      Temp5=VAL(Inp$[6;5])
3930                      IF Temp4=0 THEN
3935                         S1_sample(Solution,No)=0
3940                         S1_dilute(Solution,No)=0
3945                         PRINT TABXY(12,No+14);RPT$(" ",16)
3950                         No=(No MOD 4)+1
3955                      ELSE
3960                      IF Temp4>0 AND Temp4<=10 THEN
3965                         IF Temp5>=0 AND Temp5<=200 THEN
3970                            S1_sample(Solution,No)=Temp4
3975                            S1_dilute(Solution,No)=Temp5
3980                            PRINT TABXY(12,No+14);RPT$(" ",4-LEN(VAL$(Temp4)))
&VAL$(Temp4)
3985                            PRINT TABXY(20,No+14);RPT$(" ",5-LEN(VAL$(Temp5)))
&VAL$(Temp5)
3990                            No=(No MOD 4)+1
3995                         ELSE
4000                            CALL Msg_disp(1)
4005                         END IF
4010                      ELSE
4015                         CALL Msg_disp(1)
4020                      END IF
4025                   END IF
4030                   ELSE
4035                      CALL Msg_disp(2)
4040                   END IF
4045                END IF
4050             END WHILE
4055             OFF ERROR
4060          CASE =2
4065             Key_no=-1
4070             No=1
4075             WHILE Key_no<0 OR (Key_no)3 AND Key_no<9)
4080 S2_input:!
4085                ON ERROR GOTO Valid_data2
4090                Guide$="Cup no.,sampling,# : "
4095                Temp1=S2_cup(Solution,No)
4100                Temp3=S2_sample(Solution,No)
4105                Temp2=S2_numbers(Solution,No)
4110                Inp$=VAL$(Temp1)&","
4115                Inp$=Inp$&RPT$(" ",4-LEN(VAL$(Temp3)))&VAL$(Temp3)&","
4120                Inp$=Inp$&RPT$(" ",2-LEN(VAL$(Temp2)))&VAL$(Temp2)
4125                PRINT TABXY(32+25*((No-1) DIV 10),((No-1) MOD 10)+9);"*";
4130                CALL Key_in(3,Inp$,Guide$,Key_no)
4135                PRINT TABXY(32+25*((No-1) DIV 10),((No-1) MOD 10)+9);" ";
4140                IF Key_no<0 THEN
4145                   IF POS(Inp$,",")=2 AND POS(Inp$[3],",")=5 THEN
4150                      Temp1=VAL(Inp$[1;1])
4155                      Temp3=INT(100*VAL(Inp$[3;4])+.5)/100
4160                      Temp2=VAL(Inp$[8;2])
4161                      IF Temp1=0 THEN
4162                         S2_cup(Solution,No)=0
4163                         S2_sample(Solution,No)=0
4164                         S2_numbers(Solution,No)=0
4165                         X=25*((No-1) DIV 10)
4166                         Y=((No-1) MOD 10)+9
4167                         PRINT TABXY(X+36,Y);RPT$(" ",17)
4168                         No=(No MOD 20)+1
4169                      ELSE
```

```
4171                        IF Temp1>0 AND Temp1<5 THEN
4172                           IF Temp2>0 AND Temp2<21 THEN
4175                              IF Temp3>=0 AND Temp3<=10 THEN
4180                                 S2_cup(Solution,No)=Temp1
4185                                 S2_sample(Solution,No)=Temp3
4190                                 S2_numbers(Solution,No)=Temp2
4195                                 X=25*((No-1) DIV 10)
4200                                 Y=((No-1) MOD 10)+9
4205                                 PRINT TABXY(X+36,Y);VAL$(Temp1)
4210                                 PRINT TABXY(X+41,Y);RPT$(" ",4-LEN(VAL$(Temp3))
)&VAL$(Temp3)
4215                                 PRINT TABXY(X+51,Y);RPT$(" ",2-LEN(VAL$(Temp2))
)&VAL$(Temp2)
4220                                 No=(No MOD 20)+1
4225                              ELSE
4230                                 CALL Msg_disp(1)
4235                              END IF
4240                           ELSE
4245                              CALL Msg_disp(1)
4250                           END IF
4255                        ELSE
4260                           CALL Msg_disp(1)
4265                        END IF
4266                        END IF
4270                     ELSE
4275                        CALL Msg_disp(2)
4280                     END IF
4285                  END IF
4290               END WHILE
4295               OFF ERROR
4300            CASE =3
4305               CALL Set_bottle(2,Key_no)
4310               IF Key_no<>9 THEN Set_sample
4315            END SELECT
4320         END WHILE
4325         OFF ERROR
4330         SUBEXIT
4335         !
4340 Valid_data1:!
4345         CALL Msg_disp(1)
4350         OFF ERROR
4355         GOTO S1_input
4360 Valid_data2:!
4365         CALL Msg_disp(1)
4370         OFF ERROR
4375         GOTO S2_input
4380  !
4385  !----- SUBROUTINES
4390  !
4395 Frame_1:!
4400         RPLOT 0,0
4405         RPLOT 130,0
4410         RPLOT 130,-48
4415         RPLOT 0,-48
4420         RPLOT 0,0
4425         RPLOT 62,0
4430         RPLOT 62,-48
4435         RETURN
4440 Frame_2:!
4445         RPLOT 0,0
4450         RPLOT 120,0
4455         RPLOT 120,-93
4460         RPLOT 0,-93
4465         RPLOT 0,0
4470         RPLOT 0,-36
4475         RPLOT 120,-36,2
4480         RPLOT 25,0
4485         RPLOT 25,-93
4490         RPLOT 70,-93
4495         RPLOT 70,0
4500         RETURN
4505 Frame_3:!
```

```
4510      RPLOT 0,0
4515      RPLOT 110,0
4520      RPLOT 110,-165
4525      RPLOT 0,-165
4530      RPLOT 0,0
4535      RPLOT 0,-36
4540      RPLOT 110,-36,2
4545      RPLOT 25,0
4550      RPLOT 25,-165
4555      RPLOT 70,-165
4560      RPLOT 70,0
4565      RETURN
4570   SUBEND
4575   !
4580   SUB Set_bottle(INTEGER Function,Key_no)
4585   !*****************************************
4590   !*
4595   !*      Set the Bottle-No.
4600   !*
4605   !*****************************************
4610      OPTION BASE 1
4615      COM /Bottle_data/ Bottle$(*),Inp_vial(*)
4620      DIM Inp$[20],Guide$[40]
4625      DIM Bot$[6],Bottle1$[6],Bottle2$[6]
4630      REAL Vial,Inp_vial1,Inp_vial2
4635      INTEGER No
4640      !
4645 Set_bottle:!
4650      OUTPUT 2;"K";
4655      GCLEAR
4660     !Guide$="%P%$%d%k%S%s#N#o!%%G!<%?@_Dj"
4665     !CALL K_move(28,92)
4670     !CALL K_label(Guide$)
4675      PRINT TABXY(30,1);"Preset Number of Vials"
4680      MOVE 27,246
4685      GOSUB Frame_1
4690      MOVE 227,246
4695      GOSUB Frame_1
4700      PRINT TABXY(9,6);"No.";TABXY(14,6);"  Vial No.";TABXY(30,6);"Weight"
4705      PRINT TABXY(49,6);"No.";TABXY(54,6);"  Vial No.";TABXY(70,6);"Weight"
4710      FOR No=1 TO 10
4715         Bottle1$=Bottle$(No)
4720         Bottle2$=Bottle$(No+10)
4725         Inp_vial1=Inp_vial(No)
4730         Inp_vial2=Inp_vial(No+10)
4735         PRINT TABXY(8,No+7);RPT$(" ",3-LEN(VAL$(No)))&VAL$(No)
4740         PRINT TABXY(17,No+7);Bottle1$
4745         IF Inp_vial1>0 THEN
4750            PRINT TABXY(30,No+7);VAL$(Inp_vial1)
4755         END IF
4760         PRINT TABXY(48,No+7);" "&VAL$(No+10)
4765         PRINT TABXY(57,No+7);Bottle2$
4770         IF Inp_vial2>0 THEN
4775            PRINT TABXY(70,No+7);VAL$(Inp_vial2)
4780         END IF
4785      NEXT No
4790      !
4795      No=1
4800      Key_no=0
4805      WHILE Key_no<>9 AND (Function<>2 OR Key_no<>3)
4810         PRINT TABXY(5+40*((No-1) DIV 10),((No-1) MOD 10)+8);"*"
4815         Guide$="(Bottle #"&VAL$(No)&") No.,Vial_Weight : "
4820         Bot$=Bottle$(No)
4825         Vial=Inp_vial(No)
4830         Inp$=RPT$(" ",6-LEN(Bot$))&Bot$&","
4835         Inp$=Inp$&RPT$(" ",7-LEN(VAL$(Vial)))&VAL$(Vial)
4840         IF Function=1 THEN
4845            Func=5
4850         ELSE
4855            Func=4
4860         END IF
4865         CALL Key_in(Func,Inp$,Guide$,Key_no)
```

```
4870        PRINT TABXY(5+40*((No-1) DIV 10),((No-1) MOD 10)+8);" "
4875        IF Key_no<0 THEN
4880           IF POS(Inp$,",")=7 THEN
4885              Bot$=Inp$[1;6]
4890              Vial=VAL(Inp$[8;7])
4895              Bottle$(No)=Bot$
4900              Inp_vial(No)=Vial
4905              PRINT TABXY(17+40*((No-1) DIV 10),((No-1) MOD 10)+8);Bot$
4910              IF Vial>0 THEN
4915                 PRINT TABXY(29+40*((No-1) DIV 10),((No-1) MOD 10)+8);Vial
4920              ELSE
4925                 PRINT TABXY(29+40*((No-1) DIV 10),((No-1) MOD 10)+8);RPT$("
 ",8)
4930              END IF
4935              No=(No MOD 20)+1
4940           END IF
4945        END IF
4950     END WHILE
4955     SUBEXIT
4960  !
4965  !----- SUBROUTINES
4970  !
4975 Frame_1:!
4980     RPLOT 0,0
4985     RPLOT 165,0
4990     RPLOT 165,-154
4995     RPLOT 0,-154
5000     RPLOT 0,0
5005     RPLOT 0,-24
5010     RPLOT 165,-24,2
5015     RPLOT 35,0
5020     RPLOT 35,-154,2
5025     RPLOT 100,0
5030     RPLOT 100,-154
5035     RETURN
5040  SUBEND
5045  !
5050  SUB Print_sample(INTEGER Solution,Ret_code)
5055  !************************************************
5060  !*
5065  !*      Print  Data of Sampling
5070  !*
5075  !************************************************
5080     OPTION BASE 1
5085     COM /Chit_data/ Unit$(*),Chit$(*),Name$(*),Date$(*)
5090     COM /Sample_1/ REAL S1_sample(*),S1_dilute(*)
5095     COM /Sample_2/ INTEGER S2_cup(*),S2_numbers(*)
5100     COM /Sample_2/ REAL S2_sample(*)
5105     DIM Buf$[24],Buf1$[32],Buf2$[24],Buf3$[24]
5110     INTEGER I,J,K
5115     INTEGER Temp1,Temp2
5120     REAL Temp3,Temp4,Temp5
5125     !
5130 Print_sample:!
5135     PRINTER IS 30;WIDTH 300
5140     PRINT CHR$(27)&"C"&CHR$(6);
5145     PRINT CHR$(27)&"I"&CHR$(136);
5150     PRINT CHR$(27)&"8"&CHR$(0)&CHR$(0);
5155     PRINT
5160     PRINT
5165     PRINT
5170     PRINT "!!!!!!!!!!!!!!!!!!!!!!!!";
5175     IF Solution=1 THEN
5180       !PRINT "!!!!!!!!!!861U!J#A!K7WB,MQ%G!<%?@_Dj"
5185        PRINT "#P#r#e#s#e#t!!#D#a#t#a!!#o#f!!#D#i#l#u#t#i#o#n!!#A"
5190     ELSE
5195       !PRINT "!!!!!!!!!!861U!J#B!K7WB,MQ%G!<%?@_Dj"
5200        PRINT "#P#r#e#s#e#t!!#D#a#t#a!!#o#f!!#D#i#l#u#t#i#o#n!!#B"
5205     END IF
5210     PRINT
5215     PRINT
5220     PRINT
```

```
5225      PRINT
5230      PRINT
5235      PRINT
5240      PRINT
5245      PRINT CHR$(27)&"I"&CHR$(136);
5250      PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4);
5255      PRINT CHR$(27)&"C"&CHR$(12);
5260      FOR J=1 TO 7
5265         GOSUB L_margin
5270         SELECT J
5275         CASE 1
5280            PRINT RPT$(" ",35);
5285            GOSUB Line_31
5290         CASE 2,4,6
5295            PRINT RPT$(" ",35);
5300            GOSUB Line_32
5305         CASE 3
5310            PRINT RPT$(" ",35);
5315            GOSUB Line_menu21
5320         CASE 5
5325            PRINT RPT$(" ",35);
5330            GOSUB Line_menu22
5335         CASE 7
5340            GOSUB Line_11
5345            GOSUB Line_33
5350         END SELECT
5355         PRINT
5360      NEXT J
5365      FOR I=1 TO 4
5370         FOR J=1 TO 4
5375            GOSUB L_margin
5380            SELECT J
5385            CASE 1,3
5390               IF I=4 THEN
5395                  PRINT RPT$(" ",35);
5400               ELSE
5405                  GOSUB Line_12
5410               END IF
5415               GOSUB Line_32
5420            CASE 2
5425               SELECT I
5430               CASE 1
5435                  Buf$="   "&Unit$(Solution)&"   "
5440                  CALL Kanji_conv(Buf$,Buf1$)
5445                  GOSUB Line_data11
5450               CASE 2
5455                  Buf$="   "&Chit$(Solution)&"     "
5460                  CALL Kanji_conv(Buf$,Buf1$)
5465                  GOSUB Line_data12
5470               CASE 3
5475                  Buf$="   "&Name$(Solution)&"   "
5480                  CALL Kanji_conv(Buf$,Buf1$)
5485                  GOSUB Line_data13
5490               CASE 4
5495                  PRINT RPT$(" ",35);
5500               END SELECT
5505               Buf$=RPT$(" ",4-LEN(VAL$(I)))&VAL$(I)&"  "
5510               CALL Kanji_conv(Buf$,Buf1$)
5515               Temp4=S1_sample(Solution,I)
5520               IF Temp4=0 THEN
5525                  Buf2$=RPT$("/'",12)
5530                  Buf3$=RPT$("/'",12)
5535               ELSE
5540                  Buf$=RPT$(" ",8-LEN(VAL$(Temp4)))&VAL$(Temp4)&"    "
5545                  CALL Kanji_conv(Buf$,Buf2$)
5550                  Temp5=S1_dilute(Solution,I)
5555                  Buf$=RPT$(" ",8-LEN(VAL$(Temp5)))&VAL$(Temp5)&"    "
5560                  CALL Kanji_conv(Buf$,Buf3$)
5565               END IF
5570               GOSUB Line_data2
5575            CASE 4
5580               IF I=4 THEN
```

```
5585                PRINT RPT$(" ",35);
5590                GOSUB Line_34
5595             ELSE
5600                IF I=3 THEN
5605                   GOSUB Line_14
5610                ELSE
5615                   GOSUB Line_13
5620                END IF
5625                GOSUB Line_33
5630             END IF
5635          END SELECT
5640          PRINT
5645       NEXT J
5650    NEXT I
5655    PRINT
5660    PRINT
5665    PRINT
5670    FOR J=1 TO 7
5675       GOSUB L_margin
5680       FOR K=1 TO 2
5685          SELECT J
5690          CASE 1
5695             GOSUB Line_31
5700          CASE 2,4,6
5705             GOSUB Line_32
5710          CASE 3
5715             GOSUB Line_menu31
5720          CASE 5
5725             GOSUB Line_menu32
5730          CASE 7
5735             GOSUB Line_33
5740          END SELECT
5745       NEXT K
5750       PRINT
5755    NEXT J
5760    FOR I=1 TO 10
5765       FOR J=1 TO 4
5770          GOSUB L_margin
5775          FOR K=1 TO 2
5780             SELECT J
5785             CASE 1,3
5790                GOSUB Line_32
5795             CASE 2
5800                Temp1=S2_cup(Solution,I+10*(K-1))
5805                IF Temp1=0 THEN
5810                   Buf1$=RPT$("/",6)
5815                   Buf2$=RPT$("/",12)
5820                   Buf3$=RPT$("/",12)
5825                ELSE
5830                   Buf$=RPT$(" ",4-LEN(VAL$(Temp1)))&VAL$(Temp1)&"   "
5835                   CALL Kanji_conv(Buf$,Buf1$)
5840                   Temp3=S2_sample(Solution,I+10*(K-1))
5845                   Buf$=RPT$(" ",8-LEN(VAL$(Temp3)))&VAL$(Temp3)&"     "
5850                   CALL Kanji_conv(Buf$,Buf2$)
5855                   Temp2=S2_numbers(Solution,I+10*(K-1))
5860                   Buf$=RPT$(" ",7-LEN(VAL$(Temp2)))&VAL$(Temp2)&"      "
5865                   CALL Kanji_conv(Buf$,Buf3$)
5870                END IF
5875                GOSUB Line_data3
5880             CASE 3
5885                GOSUB Line_32
5890             CASE 4
5895                IF I=10 THEN
5900                   GOSUB Line_34
5905                ELSE
5910                   GOSUB Line_33
5915                END IF
5920             END SELECT
5925          NEXT K
5930          PRINT
5935       NEXT J
5940    NEXT I
```

```
5945      PRINT CHR$(12);
5950      PRINTER IS 1
5955      SUBEXIT
5960    !
5965    !------- SUBROUTINES
5970    !
5975  L_margin:!
5980      PRINT "!!";
5985      RETURN
5990  Line_data11:!
5995      PRINT CHR$(27)&"H"&CHR$(5);
6000      PRINT "/'"&CHR$(47)&CHR$(122);
6005      PRINT "/'/'/'.U.n.i.t/'.N.o../'/'/'/'";
6010      PRINT CHR$(47)&CHR$(122)&Buf1$;
6015      PRINT CHR$(47)&CHR$(122);
6020      RETURN
6025  Line_data12:!
6030      PRINT CHR$(27)&"H"&CHR$(5);
6035      PRINT "/'"&CHR$(47)&CHR$(122);
6040      PRINT "/'/'/'.R.e.p.o.r.t/'.N.o../'/'";
6045      PRINT CHR$(47)&CHR$(122)&Buf1$;
6050      PRINT CHR$(47)&CHR$(122);
6055      RETURN
6060  Line_data13:!
6065      PRINT CHR$(27)&"H"&CHR$(5);
6070      PRINT "/'"&CHR$(47)&CHR$(122);
6075      PRINT "/'/'/'.O.p.e.r.a.t.o.r/'/'/'/'";
6080      PRINT CHR$(47)&CHR$(122)&Buf1$;
6085      PRINT CHR$(47)&CHR$(122);
6090      RETURN
6095  Line_11:!
6100      PRINT CHR$(27)&"H"&CHR$(5);
6105      PRINT "/'"&CHR$(47)&CHR$(100)&RPT$(CHR$(47)&CHR$(99),15);
6110      PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),16);
6115      PRINT CHR$(47)&CHR$(101);
6120      RETURN
6125  Line_12:!
6130      PRINT CHR$(27)&"H"&CHR$(5);
6135      PRINT "/'"&CHR$(47)&CHR$(122)&RPT$("/'",15);
6140      PRINT CHR$(47)&CHR$(122)&RPT$("/'",16);
6145      PRINT CHR$(47)&CHR$(122);
6150      RETURN
6155  Line_13:!
6160      PRINT CHR$(27)&"H"&CHR$(5);
6165      PRINT "/'"&CHR$(47)&CHR$(107)&RPT$(CHR$(47)&CHR$(99),15);
6170      PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),16);
6175      PRINT CHR$(47)&CHR$(105);
6180      RETURN
6185  Line_14:!
6190      PRINT CHR$(27)&"H"&CHR$(5);
6195      PRINT "/'"&CHR$(47)&CHR$(103)&RPT$(CHR$(47)&CHR$(99),15);
6200      PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),16);
6205      PRINT CHR$(47)&CHR$(102);
6210      RETURN
6215  Line_menu21:!
6220      PRINT CHR$(27)&"H"&CHR$(5);
6225      PRINT "/'"&CHR$(47)&CHR$(122);
6230      PRINT "/'.C.u.p/'/'";
6235      PRINT CHR$(47)&CHR$(122);
6240      PRINT CHR$(27)&"8"&CHR$(0)&CHR$(0)&"/'/'.1.s.t/'.S.a.m.p.l.i.n.g/'/'";
6245      PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4)&CHR$(47)&CHR$(122);
6250      PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4)&"/'/'.D.i.l.u.t.i.o.n/'/'";
6255      PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4)&CHR$(47)&CHR$(122);
6260      RETURN
6265  Line_menu22:!
6270      PRINT CHR$(27)&"H"&CHR$(5);
6275      PRINT "/'"&CHR$(47)&CHR$(122);
6280      PRINT "/'/'.N.o../'";
6285      PRINT CHR$(47)&CHR$(122);
6290      PRINT "/'/'/'/'.(.m.l.)/'/'/'/'";
6295      PRINT CHR$(47)&CHR$(122);
6300      PRINT "/'/'/'/'.(.m.l.)/'/'/'/'";
```

```
6305      PRINT CHR$(47)&CHR$(122);
6310      RETURN
6315 Line_data2:!
6320      PRINT CHR$(27)&"H"&CHR$(5);
6325      PRINT "/'"&CHR$(47)&CHR$(122)&Buf1$;
6330      PRINT CHR$(47)&CHR$(122)&Buf2$;
6335      PRINT CHR$(47)&CHR$(122)&Buf3$;
6340      PRINT CHR$(47)&CHR$(122);
6345      RETURN
6350      !
6355 Line_menu31:!
6360      PRINT CHR$(27)&"H"&CHR$(5);
6365      PRINT "/'"&CHR$(47)&CHR$(122);
6370      PRINT "/'.C.u.p/'/'";
6375      PRINT CHR$(47)&CHR$(122);
6380      PRINT CHR$(27)&"8"&CHR$(0)&CHR$(0)&"/'/'.2.n.d/'.S.a.m.p.l.i.n.g/'/'";
6385      PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4)&CHR$(47)&CHR$(122);
6390      PRINT CHR$(27)&"8"&CHR$(0)&CHR$(0)&".N.u.m.b.e.r/'.o.f/'.V.i.a.l.s/'";
6395      PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4)&CHR$(47)&CHR$(122);
6400      RETURN
6405 Line_menu32:!
6410      PRINT CHR$(27)&"H"&CHR$(5);
6415      PRINT "/'"&CHR$(47)&CHR$(122);
6420      PRINT "/'/'.N.o..../'";
6425      PRINT CHR$(47)&CHR$(122);
6430      PRINT "/'/'/'/'.(.m.l.)/'/'/'/'";
6435      PRINT CHR$(47)&CHR$(122);
6440      PRINT "/'/'.(.p.i.e.c.e.s.)/'/'";
6445      PRINT CHR$(47)&CHR$(122);
6450      RETURN
6455 Line_data3:!
6460      PRINT CHR$(27)&"H"&CHR$(5);
6465      PRINT "/'"&CHR$(47)&CHR$(122)&Buf1$;
6470      PRINT CHR$(47)&CHR$(122)&Buf2$;
6475      PRINT CHR$(47)&CHR$(122)&Buf3$;
6480      PRINT CHR$(47)&CHR$(122);
6485      RETURN
6490 Line_31:!
6495      PRINT CHR$(27)&"H"&CHR$(5);
6500      PRINT "/'"&CHR$(47)&CHR$(100)&RPT$(CHR$(47)&CHR$(99),6);
6505      PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),12);
6510      PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),12);
6515      PRINT CHR$(47)&CHR$(101);
6520      RETURN
6525 Line_32:!
6530      PRINT CHR$(27)&"H"&CHR$(5);
6535      PRINT "/'"&CHR$(47)&CHR$(122)&RPT$("/'",6);
6540      PRINT CHR$(47)&CHR$(122)&RPT$("/'",12);
6545      PRINT CHR$(47)&CHR$(122)&RPT$("/'",12);
6550      PRINT CHR$(47)&CHR$(122);
6555      RETURN
6560 Line_33:!
6565      PRINT CHR$(27)&"H"&CHR$(5);
6570      PRINT "/'"&CHR$(47)&CHR$(107)&RPT$(CHR$(47)&CHR$(99),6);
6575      PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),12);
6580      PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),12);
6585      PRINT CHR$(47)&CHR$(105);
6590      RETURN
6595 Line_34:!
6600      PRINT CHR$(27)&"H"&CHR$(5);
6605      PRINT "/'"&CHR$(47)&CHR$(103)&RPT$(CHR$(47)&CHR$(99),6);
6610      PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),12);
6615      PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),12);
6620      PRINT CHR$(47)&CHR$(102);
6625      RETURN
6630  SUBEND
6635  !
6640  SUB Print_bottle(INTEGER Ret_code)
6645  !*****************************************************
6650  !*
6655  !*      Print Bottle-no.
6660  !*
```

```
6665  !************************************************
6670     OPTION BASE 1
6675     COM /Bottle_data/ Bottle$(20)[6],Inp_vial(20)
6680     DIM Buf$[48],Buf1$[48],Buf2$[48],Buf3$[48]
6685     REAL Vial
6690     INTEGER I,J,K
6695     !
6700 Print_bottle:!
6705     PRINTER IS 30;WIDTH 300
6710     PRINT CHR$(27)&"C"&CHR$(6);
6715     PRINT CHR$(27)&"I"&CHR$(136);
6720     PRINT CHR$(27)&"8"&CHR$(0)&CHR$(0);
6725     PRINT
6730     PRINT
6735     PRINT "!!!!!!!!!!!!!!!!!!!!!!!!!";
6740     PRINT "#P#r#e#s#e#t!!#N#u#m#b#e#r!!#o#f!!#V#i#a#l#s"
6745     PRINT
6750     PRINT
6755     PRINT
6760     PRINT
6765     PRINT
6770     PRINT
6775     PRINT
6780     PRINT CHR$(27)&"I"&CHR$(136);
6785     PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4);
6790     PRINT CHR$(27)&"C"&CHR$(12);
6795     FOR J=1 TO 5
6800        GOSUB L_margin
6805        FOR K=1 TO 2
6810           SELECT J
6815           CASE 1
6820              GOSUB Line_1
6825           CASE 2,4
6830              GOSUB Line_2
6835           CASE 3
6840              GOSUB Line_menu
6845           CASE 5
6850              GOSUB Line_3
6855           END SELECT
6860        NEXT K
6865        PRINT
6870     NEXT J
6875     FOR I=1 TO 10
6880        FOR J=1 TO 4
6885           GOSUB L_margin
6890           FOR K=1 TO 2
6895              SELECT J
6900              CASE 1,3
6905                 GOSUB Line_2
6910              CASE 2
6915                 Buf$=RPT$(" ",3-LEN(VAL$(I+10*(K-1))))&VAL$(I+10*(K-1))&" "
6920                 CALL Kanji_conv(Buf$,Buf1$)
6925                 Buf$=RPT$(" ",3)&Bottle$(I+10*(K-1))&RPT$(" ",3)
6930                 CALL Kanji_conv(Buf$,Buf2$)
6935                 Vial=Inp_vial(I+10*(K-1))
6940                 IF Vial>0 THEN
6945                    Buf$=RPT$(" ",4)&VAL$(Vial)&RPT$(" ",7-LEN(VAL$(Vial)))&
RPT$(" ",4)
6950                 ELSE
6955                    Buf$=RPT$(" ",15)
6960                 END IF
6965                 CALL Kanji_conv(Buf$,Buf3$)
6970                 GOSUB Line_data
6975              CASE 3
6980                 GOSUB Line_3
6985              CASE 4
6990                 IF I=10 THEN
6995                    GOSUB Line_4
7000                 ELSE
7005                    GOSUB Line_3
7010                 END IF
7015              END SELECT
```

```
7020            NEXT K
7025            PRINT
7030          NEXT J
7035       NEXT I
7040       PRINT CHR$(12);
7045       PRINTER IS 1
7050       Ret_code=0
7055       SUBEXIT
7060    !
7065    !------------- SUBROUTINES
7070    !
7075  L_margin:!
7080       PRINT "!!";
7085       RETURN
7090  Line_menu:!
7095       PRINT CHR$(27)&"H"&CHR$(5);
7100       PRINT CHR$(47)&CHR$(122)&"/'.N.o..";
7105       PRINT CHR$(47)&CHR$(122);CHR$(27)&"8"&CHR$(4)&CHR$(4);
7110       PRINT "/'/'.V.i.a.l/'.N.o../'/'"&CHR$(47)&CHR$(122);
7115       PRINT "/'/'. .W.e.i.g.h.t. /'.(.g.)/'"&CHR$(47)&CHR$(122);
7120       RETURN
7125  Line_data:!
7130       PRINT CHR$(27)&"H"&CHR$(5);
7135       PRINT CHR$(47)&CHR$(122)&Buf1$;
7140       PRINT CHR$(47)&CHR$(122)&Buf2$;
7145       PRINT CHR$(47)&CHR$(122)&Buf3$;
7150       PRINT CHR$(47)&CHR$(122);
7155       RETURN
7160  Line_1:!
7165       PRINT CHR$(27)&"H"&CHR$(5);
7170       PRINT CHR$(47)&CHR$(100)&RPT$(CHR$(47)&CHR$(99),4);
7175       PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),12);
7180       PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),15);
7185       PRINT CHR$(47)&CHR$(101);
7190       RETURN
7195  Line_2:!
7200       PRINT CHR$(27)&"H"&CHR$(5);
7205       PRINT CHR$(47)&CHR$(122)&RPT$("/'",4);
7210       PRINT CHR$(47)&CHR$(122)&RPT$("/'",12);
7215       PRINT CHR$(47)&CHR$(122)&RPT$("/'",15);
7220       PRINT CHR$(47)&CHR$(122);
7225       RETURN
7230  Line_3:!
7235       PRINT CHR$(27)&"H"&CHR$(5);
7240       PRINT CHR$(47)&CHR$(107)&RPT$(CHR$(47)&CHR$(99),4);
7245       PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),12);
7250       PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),15);
7255       PRINT CHR$(47)&CHR$(105);
7260       RETURN
7265  Line_4:!
7270       PRINT CHR$(27)&"H"&CHR$(5);
7275       PRINT CHR$(47)&CHR$(103)&RPT$(CHR$(47)&CHR$(99),4);
7280       PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),12);
7285       PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),15);
7290       PRINT CHR$(47)&CHR$(102);
7295       RETURN
7300    SUBEND
7305    !
7310    SUB State_samp(INTEGER Ret_code)
7315    !************************************************
7320    !*
7325    !*    1st,Dilution,2nd  Sampling
7330    !*
7335    !************************************************
7340       OPTION BASE 1
7345       COM /Sample_1/ REAL S1_sample(*),S1_dilute(*)
7350       COM /Sample_2/ INTEGER S2_cup(*),S2_numbers(*),REAL S2_sample(*)
7355       COM /Result/ REAL Sample_1st(*),Dilution(*)
7360       COM /Result/ REAL Sample_2nd(*),INTEGER Cup_no(*),REAL Mea_vial(*)
7365       COM /Measure/ INTEGER Solution,Sampling,Max_cup,Max_bottle
7370       COM /Measure/ INTEGER Now_cup,Now_bottle,S2_point
7375       DIM Buf$[50],Guide$[32]
```

```
7380      INTEGER I,J,N,L,Key_no,Flag1,No
7385      REAL Data,Temp
7390      REAL Fuutai_c(4),S_1ji(4),Dil(4),Fuutai_b(20),S_2ji(20)
7395      REAL Dfuutai_c,Ds_1ji,Ddil1,Dfuutai_b,Ds_2ji
7400      REAL Sample_x,Max_vol
7405      !
7410 State_samp:!
7415      Ret_code=0
7420      OFF KEY
7425      ON KNOB .1 CALL Dummy
7430      ON KEY 9 LABEL "(END)" CALL Dummy
7435      ON KEY 19 LABEL "(END)",3 RECOVER End_key
7440      !
7445      CALL Move_sampler(0,0,Ret_code)
7450      IF Ret_code<>0 THEN GOTO End_key
7455      CALL Move_arm(0,0,Ret_code)
7460      IF Ret_code<>0 THEN GOTO End_key
7465      CALL Cover(0,Ret_code)
7470      IF Ret_code<>0 THEN GOTO End_key
7475      CALL C_table(0,Ret_code)
7480      IF Ret_code<>0 THEN GOTO End_key
7485      CALL B_table(0,Ret_code)
7490      IF Ret_code<>0 THEN GOTO End_key
7495      CALL Tenbin(0,0,Ret_code)
7500      IF Ret_code<>0 THEN GOTO End_key
7505      !
7510      Now_cup=0
7515      Now_bottle=0
7520      FOR I=1 TO 2
7525         FOR J=1 TO 20
7530            Cup_no(I,J)=0
7535            Sample_2nd(I,J)=0
7540         NEXT J
7545      NEXT I
7550      FOR I=1 TO 20
7555         Mea_vial(I)=0
7560      NEXT I
7565      FOR I=1 TO 2
7570         FOR J=1 TO 4
7575            Sample_1st(I,J)=0
7580            Dilution(I,J)=0
7585         NEXT J
7590      NEXT I
7595      FOR Solution=1 TO 2
7600         Flag1=0
7605         FOR I=1 TO 4
7610            IF S1_sample(Solution,I)>0 THEN Flag1=1
7615         NEXT I
7620         IF Flag1>0 THEN
7625            GOSUB Chip_set
7630            FOR I=1 TO 4
7635               IF S1_sample(Solution,I)>0 THEN
7640                  Sampling=1
7645                  Now_cup=I
7650                  Max_cup=Now_cup
7655                  CALL Disp_measure(0)
7660                  CALL Disp_measure(15)
7665                  CALL Disp_measure(40)
7670                  CALL C_table(Now_cup,Ret_code)
7675                  IF Ret_code<>0 THEN GOTO End_key
7680                  GOSUB Pcup_set
7685                  CALL Cover(0,Ret_code)
7690                  IF Ret_code<>0 THEN GOTO End_key
7695                  CALL Tenbin(1,Data,Ret_code)
7700                  IF Ret_code<>0 THEN GOTO End_key
7705                  IF Data>5 THEN
7710                     Fuutai_c(Now_cup)=Data
7715                  ELSE
7720                     Ret_code=9
7725                     GOTO End_key
7730                  END IF
7735                  GOSUB Samp_cup
```

```
7740                CALL Cover(0,Ret_code)
7745                IF Ret_code<>0 THEN GOTO End_key
7750                CALL Tenbin(1,Data,Ret_code)
7755                IF Ret_code<>0 THEN GOTO End_key
7760                IF Data>S THEN
7765                   S_1ji(Now_cup)=Data
7770                ELSE
7775                   Ret_code=9
7780                   GOTO End_key
7785                END IF
7790                Sample_1st(Solution,Now_cup)=S_1ji(Now_cup)-Fuutai_c(Now_cu
p)
7795                CALL Disp_measure(15)
7800                CALL Move_arm(1,0,Ret_code)
7805                IF Ret_code<>0 THEN GOTO End_key
7810                CALL Cover(3,Ret_code)
7815                IF Ret_code<>0 THEN GOTO End_key
7820                CALL Move_arm(6,5,Ret_code)
7825                IF Ret_code<>0 THEN GOTO End_key
7830                CALL Move_hand(0,7,Ret_code)
7835                IF Ret_code<>0 THEN GOTO End_key
7840                CALL Move_arm(1,0,Ret_code)
7845                IF Ret_code<>0 THEN GOTO End_key
7850                GOSUB Pcup_return
7855                CALL Move_arm(1,0,Ret_code)
7860                IF Ret_code<>0 THEN GOTO End_key
7865             END IF
7870          NEXT I
7875          CALL Cover(2,Ret_code)
7880          IF Ret_code<>0 THEN GOTO End_key
7885          GOSUB Chip_reset
7890          Sampling=2
7895          CALL Disp_measure(0)
7900          CALL Disp_measure(15)
7905          CALL Disp_measure(40)
7910          GOSUB Dilu_set
7915          FOR I=1 TO 4
7920             IF S1_dilute(Solution,I)>0 THEN
7925                Now_cup=I
7930                CALL Disp_measure(35)
7935                CALL C_table(Now_cup,Ret_code)
7940                IF Ret_code<>0 THEN GOTO End_key
7945                CALL Disp_measure(48)
7950                CALL Disp_measure(44)
7955                CALL Disp_measure(2)
7960                CALL Move_arm(2,6,Ret_code)
7965                IF Ret_code<>0 THEN GOTO End_key
7970                Temp=S1_dilute(Solution,Now_cup)
7975                CALL Titrater(1,Temp,Ret_code)
7980                IF Ret_code<>0 THEN GOTO End_key
7985                CALL Disp_measure(13)
7990                CALL Disp_measure(34)
7995                CALL Move_arm(22,7,Ret_code)
8000                IF Ret_code<>0 THEN GOTO End_key
8005             END IF
8010          NEXT I
8015          GOSUB Dilu_return
8020          FOR I=1 TO 4
8025             IF S1_dilute(Solution,I)>0 THEN
8030                J=0
8035                Now_cup=I
8040                No=I+1
8045                CALL C_table(No,Ret_code)
8050                IF Ret_code<>0 THEN GOTO End_key
8055                CALL Disp_measure(34)
8060                CALL Disp_measure(44)
8065                CALL Stir(1)
8070                CALL Disp_measure(42)
8075                WHILE J<65
8080                   WAIT 1.0
8085                   J=J+1
8090                END WHILE
```

```
8095                CALL Disp_measure(43)
8100                CALL Disp_measure(35)
8105                CALL Stir(0)
8110             END IF
8115          NEXT I
8120       END IF
8125       GOSUB Samp_vial
8130       CALL Move_arm(0,0,Ret_code)
8135       IF Ret_code<>0 THEN GOTO End_key
8140       CALL Cover(0,Ret_code)
8145       IF Ret_code<>0 THEN GOTO End_key
8150    NEXT Solution
8155    OFF KEY
8160    OFF KNOB
8165    Ret_code=0
8170    SUBEXIT
8175    ! ------------------------------------------------------------------------
8180    !    ｻﾌﾞﾙｰﾁﾝ
8185    ! ------------------------------------------------------------------------
8190 Vial_set: !  Bial-Bin no Set
8195    CALL B_table(Now_bottle,Ret_code)
8200    IF Ret_code<>0 THEN GOTO End_key
8205    IF Now_bottle>10 THEN
8210       CALL Disp_measure(24)
8215       CALL Disp_measure(45)
8220       CALL Move_hand(2,3,Ret_code)
8225       IF Ret_code<>0 THEN GOTO End_key
8230       CALL Move_arm(11,3,Ret_code)
8235       IF Ret_code<>0 THEN GOTO End_key
8240       CALL Move_hand(1,3,Ret_code)
8245       IF Ret_code<>0 THEN GOTO End_key
8250       CALL Disp_measure(4)
8255       CALL Move_arm(15,3,Ret_code)
8260       IF Ret_code<>0 THEN GOTO End_key
8265    ELSE
8270       CALL Disp_measure(28)
8275       CALL Disp_measure(45)
8280       CALL Move_arm(12,4,Ret_code)
8285       IF Ret_code<>0 THEN GOTO End_key
8290       CALL Move_hand(1,4,Ret_code)
8295       IF Ret_code<>0 THEN GOTO End_key
8300       CALL Disp_measure(6)
8305       CALL Move_arm(15,4,Ret_code)
8310       IF Ret_code<>0 THEN GOTO End_key
8315    END IF
8320    IF Now_bottle>10 THEN
8325       CALL Move_arm(8,3,Ret_code)
8330       IF Ret_code<>0 THEN GOTO End_key
8335       CALL Disp_measure(27)
8340       CALL Disp_measure(13)
8345       CALL Disp_measure(16)
8350       CALL Move_hand(0,3,Ret_code)
8355       IF Ret_code<>0 THEN GOTO End_key
8360       CALL Move_arm(15,3,Ret_code)
8365       IF Ret_code<>0 THEN GOTO End_key
8370    ELSE
8375       CALL Move_arm(8,4,Ret_code)
8380       IF Ret_code<>0 THEN GOTO End_key
8385       CALL Disp_measure(31)
8390       CALL Disp_measure(13)
8395       CALL Disp_measure(16)
8400       CALL Move_hand(0,4,Ret_code)
8405       IF Ret_code<>0 THEN GOTO End_key
8410       CALL Move_arm(15,4,Ret_code)
8415       IF Ret_code<>0 THEN GOTO End_key
8420    END IF
8425    RETURN
8430    ! ------------------------------------------------------------------------
8435 Vial_return: ! Bial-Bin no Modoshi
8440    IF Now_bottle>10 THEN
8445       CALL Move_arm(8,3,Ret_code)
8450       IF Ret_code<>0 THEN GOTO End_key
```

```
8455        CALL Move_hand(1,3,Ret_code)
8460        IF Ret_code<>0 THEN GOTO End_key
8465        CALL Disp_measure(5)
8470        CALL Move_arm(15,3,Ret_code)
8475        IF Ret_code<>0 THEN GOTO End_key
8480     ELSE
8485        CALL Move_arm(8,4,Ret_code)
8490        IF Ret_code<>0 THEN GOTO End_key
8495        CALL Move_hand(1,4,Ret_code)
8500        IF Ret_code<>0 THEN GOTO End_key
8505        CALL Disp_measure(7)
8510        CALL Move_arm(15,4,Ret_code)
8515        IF Ret_code<>0 THEN GOTO End_key
8520     END IF
8525     IF Now_bottle>10 THEN
8530        CALL Move_arm(11,3,Ret_code)
8535        IF Ret_code<>0 THEN GOTO End_key
8540        CALL Disp_measure(19)
8545        CALL Disp_measure(13)
8550        CALL Disp_measure(26)
8555        CALL Move_hand(0,3,Ret_code)
8560        IF Ret_code<>0 THEN GOTO End_key
8565     ELSE
8570        CALL Move_arm(12,4,Ret_code)
8575        IF Ret_code<>0 THEN GOTO End_key
8580        CALL Disp_measure(19)
8585        CALL Disp_measure(13)
8590        CALL Disp_measure(30)
8595        CALL Move_hand(0,4,Ret_code)
8600        IF Ret_code<>0 THEN GOTO End_key
8605     END IF
8610     IF Now_bottle>10 THEN
8615        CALL Move_arm(1,3,Ret_code)
8620        IF Ret_code<>0 THEN GOTO End_key
8625        CALL Move_hand(0,0,Ret_code)
8630        IF Ret_code<>0 THEN GOTO End_key
8635     ELSE
8640        CALL Move_arm(1,4,Ret_code)
8645        IF Ret_code<>0 THEN GOTO End_key
8650     END IF
8655     RETURN
8660     ! ------------------------------------------------------------------------
8665 Pcup_set: ! Furasuko no set
8670     IF Sampling=2 THEN
8675        CALL Disp_measure(34)
8680     ELSE
8685        CALL Disp_measure(32)
8690     END IF
8695     CALL Disp_measure(44)
8700     CALL Move_arm(10,2,Ret_code)
8705     IF Ret_code<>0 THEN GOTO End_key
8710     CALL Move_hand(1,2,Ret_code)
8715     IF Ret_code<>0 THEN GOTO End_key
8720     CALL Disp_measure(8)
8725     CALL Cover(3,Ret_code)
8730     IF Ret_code<>0 THEN GOTO End_key
8735     WAIT 15
8740     CALL Move_arm(16,2,Ret_code)
8745     IF Ret_code<>0 THEN GOTO End_key
8750     CALL Move_arm(7,2,Ret_code)
8755     IF Ret_code<>0 THEN GOTO End_key
8760     CALL Disp_measure(35)
8765     CALL Disp_measure(13)
8770     IF Sampling=2 THEN
8775        CALL Disp_measure(22)
8780     ELSE
8785        CALL Disp_measure(20)
8790     END IF
8795     CALL Move_hand(0,2,Ret_code)
8800     IF Ret_code<>0 THEN GOTO End_key
8805     CALL Move_arm(16,2,Ret_code)
8810     IF Ret_code<>0 THEN GOTO End_key
```

```
8815      RETURN
8820      ! ------------------------------------------------------------------
8825 Pcup_return: ! Furasuko no Modoshi
8830      CALL Move_arm(7,2,Ret_code)
8835      IF Ret_code<>0 THEN GOTO End_key
8840      CALL Disp_measure(9)
8845      CALL Move_hand(1,2,Ret_code)
8850      IF Ret_code<>0 THEN GOTO End_key
8855      CALL Move_arm(16,2,Ret_code)
8860      IF Ret_code<>0 THEN GOTO End_key
8865      CALL Move_arm(10,2,Ret_code)
8870      IF Ret_code<>0 THEN GOTO End_key
8875      CALL Disp_measure(23)
8880      CALL Disp_measure(13)
8885      IF Sampling=2 THEN
8890         CALL Disp_measure(34)
8895      ELSE
8900         CALL Disp_measure(48)
8905      END IF
8910      CALL Move_hand(0,2,Ret_code)
8915      IF Ret_code<>0 THEN GOTO End_key
8920      CALL Move_arm(1,0,Ret_code)
8925      IF Ret_code<>0 THEN GOTO End_key
8930      RETURN
8935      ! ------------------------------------------------------------------
8940 Chip_set: ! Chippu-saki no set
8945      CALL Move_hand(0,0,Ret_code)
8950      IF Ret_code<>0 THEN GOTO End_key
8955      CALL Move_arm(1,0,Ret_code)
8960      IF Ret_code<>0 THEN GOTO End_key
8965      CALL Move_arm(6,5,Ret_code)
8970      IF Ret_code<>0 THEN GOTO End_key
8975      CALL Move_hand(1,7,Ret_code)
8980      IF Ret_code<>0 THEN GOTO End_key
8985      CALL Move_arm(13,5,Ret_code)
8990      IF Ret_code<>0 THEN GOTO End_key
8995      CALL Move_arm(6,5,Ret_code)
9000      IF Ret_code<>0 THEN GOTO End_key
9005      CALL Move_hand(0,7,Ret_code)
9010      IF Ret_code<>0 THEN GOTO End_key
9015      CALL Move_arm(1,0,Ret_code)
9020      IF Ret_code<>0 THEN GOTO End_key
9025      RETURN
9030      ! ------------------------------------------------------------------
9035 Chip_reset: ! Chippu-saki no reset
9040      CALL Move_arm(1,0,Ret_code)
9045      IF Ret_code<>0 THEN GOTO End_key
9050      CALL Move_arm(6,5,Ret_code)
9055      IF Ret_code<>0 THEN GOTO End_key
9060      CALL Move_hand(1,7,Ret_code)
9065      IF Ret_code<>0 THEN GOTO End_key
9070      CALL Move_arm(14,5,Ret_code)
9075      IF Ret_code<>0 THEN GOTO End_key
9080      CALL Move_hand(0,7,Ret_code)
9085      IF Ret_code<>0 THEN GOTO End_key
9090      CALL Move_arm(1,0,Ret_code)
9095      IF Ret_code<>0 THEN GOTO End_key
9100      RETURN
9105      ! ------------------------------------------------------------------
9110 Samp_cup: ! Gen-eki no Sampling to C_Cup heno Hakidashi
9115      CALL Disp_measure(36)
9120      CALL Cover(3,Ret_code)
9125      IF Ret_code<>0 THEN GOTO End_key
9130      CALL Move_arm(1,0,Ret_code)
9135      IF Ret_code<>0 THEN GOTO End_key
9140      CALL Move_arm(6,5,Ret_code)
9145      IF Ret_code<>0 THEN GOTO End_key
9150      CALL Move_hand(1,7,Ret_code)
9155      IF Ret_code<>0 THEN GOTO End_key
9160      CALL Move_arm(1,0,Ret_code)
9165      IF Ret_code<>0 THEN GOTO End_key
9170      CALL Move_hand(3,7,Ret_code)
```

```
9175      IF Ret_code<>0 THEN GOTO End_key
9180      !
9185      Sample_x=S1_sample(Solution,Now_cup)
9190      WHILE Sample_x>0
9195         CALL Move_sampler(0,0,Ret_code)
9200         IF Ret_code<>0 THEN GOTO End_key
9205         CALL Move_sampler(2,0,Ret_code)
9210         IF Ret_code<>0 THEN GOTO End_key
9215         J=3+Solution
9220         CALL Move_arm(5,9,Ret_code)
9225         IF Ret_code<>0 THEN GOTO End_key
9230         CALL Move_arm(J,9,Ret_code)
9235         IF Ret_code<>0 THEN GOTO End_key
9240         CALL Disp_measure(1)
9245         CALL Move_arm(2,9,Ret_code)
9250         IF Ret_code<>0 THEN GOTO End_key
9255         IF Sample_x>4 THEN
9260            Temp=4
9265            Sample_x=Sample_x-Temp
9270         ELSE
9275            Temp=Sample_x
9280            Sample_x=0
9285         END IF
9290         CALL Move_sampler(1,Temp,Ret_code)
9295         IF Ret_code<>0 THEN GOTO End_key
9300         J=3+Solution
9305         CALL Move_arm(J,9,Ret_code)
9310         IF Ret_code<>0 THEN GOTO End_key
9315         CALL Move_sampler(1,.10,Ret_code)
9320         IF Ret_code<>0 THEN GOTO End_key
9325         CALL Move_arm(5,9,Ret_code)
9330         IF Ret_code<>0 THEN GOTO End_key
9335         CALL Move_arm(1,0,Ret_code)
9340         IF Ret_code<>0 THEN GOTO End_key
9345         CALL Move_arm(20,8,Ret_code)
9350         IF Ret_code<>0 THEN GOTO End_key
9355         CALL Move_arm(2,3,Ret_code)
9360         IF Ret_code<>0 THEN GOTO End_key
9365         CALL Move_sampler(0,0,Ret_code)
9370         IF Ret_code<>0 THEN GOTO End_key
9375         CALL Disp_measure(13)
9380         CALL Disp_measure(46)
9385         CALL Move_arm(20,8,Ret_code)
9390         IF Ret_code<>0 THEN GOTO End_key
9395         CALL Move_arm(1,0,Ret_code)
9400         IF Ret_code<>0 THEN GOTO End_key
9405      END WHILE
9410      RETURN
9415      ! ------------------------------------------------------------------------
9420 Samp_vial: ! Kishaku-ryou no Sokutei to 2Ji Sampling
9425      FOR I=1 TO 4
9430         Sampling=2
9435         Now_cup=I
9440         Max_cup=Now_cup
9445         CALL C_table(Now_cup,Ret_code)
9450         IF Ret_code<>0 THEN GOTO End_key
9455         IF S1_sample(Solution,I)>0 THEN
9460            CALL Disp_measure(0)
9465            CALL Disp_measure(15)
9470            CALL Disp_measure(40)
9475            GOSUB Pcup_set
9480            CALL Cover(0,Ret_code)
9485            IF Ret_code<>0 THEN GOTO End_key
9490            CALL Tenbin(1,Data,Ret_code)
9495            IF Ret_code<>0 THEN GOTO End_key
9500            IF Data>5 THEN
9505               Dil(Now_cup)=Data
9510            ELSE
9515               Ret_code=9
9520               GOTO End_key
9525            END IF
9530            Dilution(Solution,Now_cup)=Dil(Now_cup)-Fuutai_c(Now_cup)
```

```
9535            CALL Disp_measure(15)
9540            CALL Cover(3,Ret_code)
9545            IF Ret_code<>0 THEN GOTO End_key
9550            WAIT 15
9555            GOSUB Pcup_return
9560         END IF
9565         Max_vol=0
9570         FOR S2_point=1 TO 20
9575            IF S2_cup(Solution,S2_point)=Now_cup THEN
9580               Max_vol=Max_vol+S2_sample(Solution,S2_point)
9585            END IF
9590         NEXT S2_point
9595         IF Max_vol>0 THEN
9600            GOSUB Chip_set
9605            FOR S2_point=1 TO 20
9610               IF S2_cup(Solution,S2_point)=Now_cup THEN
9615                  FOR N=1 TO S2_numbers(Solution,S2_point)
9620                     Sampling=3
9625                     Now_bottle=Now_bottle+1
9630                     Max_bottle=Now_bottle
9635                     CALL B_table(Now_bottle,Ret_code)
9640                     IF Ret_code<>0 THEN GOTO End_key
9645                     IF S2_sample(Solution,S2_point)>0 THEN
9650                        CALL Disp_measure(0)
9655                        CALL Disp_measure(15)
9660                        CALL Disp_measure(38)
9665                        CALL Disp_measure(40)
9670                        CALL Disp_measure(34)
9675                        CALL Disp_measure(44)
9680                        GOSUB Vial_set
9685                        CALL Cover(0,Ret_code)
9690                        IF Ret_code<>0 THEN GOTO End_key
9695                        CALL Tenbin(1,Data,Ret_code)
9700                        IF Ret_code<>0 THEN GOTO End_key
9705                        IF Data>5 THEN
9710                           Fuutai_b(Now_bottle)=Data
9715                        ELSE
9720                           Ret_code=9
9725                           GOTO End_key
9730                        END IF
9735                        Mea_vial(Now_bottle)=Fuutai_b(Now_bottle)
9740                        CALL Cover(3,Ret_code)
9745                        IF Ret_code<>0 THEN GOTO End_key
9750                        CALL Move_arm(1,0,Ret_code)
9755                        IF Ret_code<>0 THEN GOTO End_key
9760                        CALL Move_arm(6,5,Ret_code)
9765                        IF Ret_code<>0 THEN GOTO End_key
9770                        CALL Move_hand(1,7,Ret_code)
9775                        IF Ret_code<>0 THEN GOTO End_key
9780                        CALL Move_arm(1,0,Ret_code)
9785                        IF Ret_code<>0 THEN GOTO End_key
9790                        CALL Move_hand(3,7,Ret_code)
9795                        IF Ret_code<>0 THEN GOTO End_key
9800                        !
9805                        Sample_x=S2_sample(Solution,S2_point)
9810                        WHILE Sample_x>0
9815                           CALL Move_sampler(0,0,Ret_code)
9820                           IF Ret_code<>0 THEN GOTO End_key
9825                           CALL Move_sampler(2,0,Ret_code)
9830                           IF Ret_code<>0 THEN GOTO End_key
9835                           CALL Move_arm(1,0,Ret_code)
9840                           IF Ret_code<>0 THEN GOTO End_key
9845                           CALL Move_arm(19,7,Ret_code)
9850                           IF Ret_code<>0 THEN GOTO End_key
9855                           CALL Move_arm(2,4,Ret_code)
9860                           IF Ret_code<>0 THEN GOTO End_key
9865                           IF Sample_x>4 THEN
9870                              Temp=4
9875                              Sample_x=Sample_x-Temp
9880                           ELSE
9885                              Temp=Sample_x
9890                              Sample_x=0
```

```
9895                            END IF
9900                            CALL Move_sampler(1,Temp,Ret_code)
9905                            IF Ret_code<>0 THEN GOTO End_key
9910                            CALL Disp_measure(3)
9915                            CALL Move_arm(19,7,Ret_code)
9920                            IF Ret_code<>0 THEN GOTO End_key
9925                            CALL Move_sampler(1,.10,Ret_code)
9930                            IF Ret_code<>0 THEN GOTO End_key
9935                            CALL Move_arm(1,0,Ret_code)
9940                            IF Ret_code<>0 THEN GOTO End_key
9945                            CALL Move_arm(17,3,Ret_code)
9950                            IF Ret_code<>0 THEN GOTO End_key
9955                            CALL Move_arm(2,17,Ret_code)
9960                            IF Ret_code<>0 THEN GOTO End_key
9965                            CALL Disp_measure(13)
9970                            CALL Disp_measure(18)
9975                            CALL Move_sampler(0,0,Ret_code)
9980                            IF Ret_code<>0 THEN GOTO End_key
9985                            CALL Move_arm(17,3,Ret_code)
9990                            IF Ret_code<>0 THEN GOTO End_key
9995                            CALL Move_arm(1,0,Ret_code)
10000                           IF Ret_code<>0 THEN GOTO End_key
10005                       END WHILE
10010                       !
10015                       CALL Cover(0,Ret_code)
10020                       IF Ret_code<>0 THEN GOTO End_key
10025                       CALL Tenbin(1,Data,Ret_code)
10030                       IF Ret_code<>0 THEN GOTO End_key
10035                       IF Data>5 THEN
10040                          S_2ji(Now_bottle)=Data
10045                       ELSE
10050                          Ret_code=9
10055                          GOTO End_key
10060                       END IF
10065                       Cup_no(Solution,Now_bottle)=Now_cup
10070                       Sample_2nd(Solution,Now_bottle)=S_2ji(Now_bottle)-Fuu
tai_b(Now_bottle)
10075                       CALL Disp_measure(15)
10080                       CALL Cover(3,Ret_code)
10085                       IF Ret_code<>0 THEN GOTO End_key
10090                       CALL Move_arm(6,5,Ret_code)
10095                       IF Ret_code<>0 THEN GOTO End_key
10100                       CALL Move_hand(0,7,Ret_code)
10105                       IF Ret_code<>0 THEN GOTO End_key
10110                       CALL Move_arm(1,0,Ret_code)
10115                       IF Ret_code<>0 THEN GOTO End_key
10120                       GOSUB Vial_return
10125                       CALL Move_arm(1,0,Ret_code)
10130                       IF Ret_code<>0 THEN GOTO End_key
10135                   END IF
10140                NEXT N
10145             END IF
10150          NEXT S2_point
10155          GOSUB Chip_reset
10160       END IF
10165    NEXT I
10170    RETURN
10175    ! ------------------------------------------------------------------------
10180 Dilu_set:! Kishaku-Chippu no set
10185    CALL Move_arm(1,0,Ret_code)
10190    IF Ret_code<>0 THEN GOTO End_key
10195    CALL Move_arm(9,6,Ret_code)
10200    IF Ret_code<>0 THEN GOTO End_key
10205    CALL Move_hand(1,6,Ret_code)
10210    IF Ret_code<>0 THEN GOTO End_key
10215    CALL Move_arm(1,6,Ret_code)
10220    IF Ret_code<>0 THEN GOTO End_key
10225    CALL Move_hand(3,6,Ret_code)
10230    IF Ret_code<>0 THEN GOTO End_key
10235    CALL Move_arm(22,7,Ret_code)
10240    IF Ret_code<>0 THEN GOTO End_key
10245    RETURN
```

```
10250    ! ------------------------------------------------------------------------
10255 Dilu_return:! Kishaku-Chippu no Modoshi
10260    CALL Move_arm(1,6,Ret_code)
10265    IF Ret_code<>0 THEN GOTO End_key
10270    CALL Move_arm(9,6,Ret_code)
10275    IF Ret_code<>0 THEN GOTO End_key
10280    CALL Move_hand(0,6,Ret_code)
10285    IF Ret_code<>0 THEN GOTO End_key
10290    CALL Move_arm(1,0,Ret_code)
10295    IF Ret_code<>0 THEN GOTO End_key
10300    RETURN
10305    ! ------------------------------------------------------------------------
10310 End_key:     !
10315    IF Ret_code=0 THEN Ret_code=9
10320    CALL Motor_1(7,0,J)
10325    CALL Motor_2(7,0,J)
10330    CALL Motor_3(7,0,J)
10335    CALL Motor_4(7,0,J)
10340    CALL Motor_5(7,0,J)
10345    OFF KEY
10350    OFF KNOB
10355    OFF TIMEOUT
10360    OFF ERROR
10365    SUBEND
10370    !
10375 SUB Status(INTEGER Mode,Ret_code)
10380 !************************************************
10385 !*
10390 !*    Check the Status
10395 !*
10400 !************************************************
10405    OPTION BASE 1
10410    INTEGER I
10415    REAL Data
10420    !
10425 Status:!
10430    Ret_code=0
10435    SELECT Mode
10440    CASE =0
10445       CALL Gmax_di(0,0,I)
10450       IF I=0 THEN Ret_code=17
10455    CASE =1
10460       CALL Tenbin(1,Data,Ret_code)
10465       IF Ret_code=0 AND Data>3 THEN Ret_code=8
10470    CASE =2
10475       CALL Tenbin(1,Data,Ret_code)
10480       IF Ret_code=0 AND Data<3 THEN Ret_code=8
10485    CASE =3
10490       CALL Tenbin(1,Data,Ret_code)
10495       IF Ret_code=0 AND Data>3 THEN Ret_code=8
10500    CASE =4
10505       CALL Tenbin(1,Data,Ret_code)
10510       IF Ret_code=0 AND Data<3 THEN Ret_code=8
10515    CASE =5
10520       CALL Tenbin(1,Data,Ret_code)
10525       IF Ret_code=0 AND Data<3 THEN Ret_code=8
10530    CASE =6
10535       CALL Tenbin(1,Data,Ret_code)
10540       IF Ret_code=0 AND Data<3 THEN Ret_code=8
10545    END SELECT
10550 SUBEND
10555 !
10560 SUB C_table(INTEGER Position,Ret_code)
10565 !************************************************
10570 !*
10575 !*     Positioning the Cup-Table
10580 !*
10585 !************************************************
10590    OPTION BASE 1
10595    COM /T_table/ INTEGER C_posi,B_posi
10600    INTEGER I,N,Senser
10605    !
```

```
10610 C_table:!
10615    SELECT Position
10620    CASE <0
10625       ON TIME (TIMEDATE+10.0) MOD 86400 RECOVER Timeout
10630       CALL Gmax_sb(0,0)
10635       LOOP
10640          CALL Gmax_di(0,4,Senser)
10645       EXIT IF Senser=0
10650       END LOOP
10655       WAIT .2
10660       CALL Gmax_rb(0,0)
10665       LOOP
10670          CALL Gmax_di(0,4,Senser)
10675       EXIT IF Senser>0
10680       END LOOP
10685       OFF TIME
10690       C_posi=(C_posi MOD 4)+1
10695       Ret_code=0
10700    CASE 0
10705       ON TIME (TIMEDATE+30.0) MOD 86400 RECOVER Timeout
10710       LOOP
10715          CALL Gmax_sb(0,0)
10720          LOOP
10725             CALL Gmax_di(0,4,Senser)
10730          EXIT IF Senser=0
10735          END LOOP
10740          WAIT .2
10745          CALL Gmax_rb(0,0)
10750          LOOP
10755             CALL Gmax_di(0,4,Senser)
10760          EXIT IF Senser>0
10765          END LOOP
10770          CALL Gmax_di(0,3,Senser)
10775       EXIT IF Senser>0
10780       END LOOP
10785       OFF TIME
10790       C_posi=1
10795    CASE ELSE
10800       IF C_posi<=Position THEN
10805          N=Position-C_posi
10810       ELSE
10815          N=4-(C_posi-Position)
10820       END IF
10825       FOR I=1 TO N
10830          ON TIME (TIMEDATE+8.0) MOD 86400 RECOVER Timeout
10835          CALL Gmax_sb(0,0)
10840          LOOP
10845             CALL Gmax_di(0,4,Senser)
10850          EXIT IF Senser=0
10855          END LOOP
10860          WAIT .2
10865          CALL Gmax_rb(0,0)
10870          LOOP
10875             CALL Gmax_di(0,4,Senser)
10880          EXIT IF Senser>0
10885          END LOOP
10890          OFF TIME
10895       NEXT I
10900       C_posi=Position
10905    END SELECT
10910    IF Position>=0 THEN
10915       WAIT 1.0
10920    !  CALL Gmax_di(0,5,Senser)
10925       SELECT Position
10930       CASE 0
10935          IF Senser>0 THEN
10940             Ret_code=0
10945          ELSE
10950             Ret_code=7
10955          END IF
10960       CASE ELSE
10965          Ret_code=0
```

```
10970       END SELECT
10975    END IF
10980    SUBEXIT
10985 Timeout:!
10990    OFF TIME
10995    CALL Gmax_rb(0,0)
11000    Ret_code=8
11005 SUBEND
11010 !
11015 SUB B_table(INTEGER Position,Ret_code)
11020 !*************************************************
11025 !*
11030 !*    Positioning the  Bottle-Table
11035 !*
11040 !*************************************************
11045    OPTION BASE 1
11050    COM /T_table/ INTEGER C_posi,B_posi
11055    INTEGER I,N,Posi,Posi_x,Senser
11060    !
11065 B_table:!
11070    SELECT Position
11075    CASE 0
11080       ON TIME (TIMEDATE+30.0) MOD 86400 RECOVER Timeout
11085       LOOP
11090          CALL Gmax_sb(0,2)
11095          LOOP
11100             CALL Gmax_di(1,1,Senser)
11105          EXIT IF Senser=0
11110          END LOOP
11115          WAIT .2
11120          CALL Gmax_rb(0,2)
11125          LOOP
11130             CALL Gmax_di(1,1,Senser)
11135          EXIT IF Senser>0
11140          END LOOP
11145          CALL Gmax_di(1,0,Senser)
11150       EXIT IF Senser>0
11155       END LOOP
11160       OFF TIME
11165       B_posi=1
11170    CASE ELSE
11175       Posi=((Position-1) MOD 10)+1
11180       IF B_posi<=Posi THEN
11185          N=Posi-B_posi
11190       ELSE
11195          N=10-(B_posi-Posi)
11200       END IF
11205       FOR I=1 TO N
11210          ON TIME (TIMEDATE+5.0) MOD 86400 RECOVER Timeout
11215          CALL Gmax_sb(0,2)
11220          LOOP
11225             CALL Gmax_di(1,1,Senser)
11230          EXIT IF Senser=0
11235          END LOOP
11240          WAIT .2
11245          CALL Gmax_rb(0,2)
11250          LOOP
11255             CALL Gmax_di(1,1,Senser)
11260          EXIT IF Senser>0
11265          END LOOP
11270          OFF TIME
11275       NEXT I
11280       B_posi=Posi
11285    END SELECT
11290    WAIT 1.0
11295 !  IF Position>10 THEN
11300 !     CALL Gmax_di(1,3,Senser)
11305 !  ELSE
11310 !     CALL Gmax_di(1,2,Senser)
11315 !  END IF
11320    SELECT Position
11325    CASE 0
```

```
11330       IF Senser>0 THEN
11335          Ret_code=0
11340       ELSE
11345          Ret_code=6
11350       END IF
11355    CASE ELSE
11360       Ret_code=0
11365    END SELECT
11370    SUBEXIT
11375 Timeout:!
11380    OFF TIME
11385    CALL Gmax_rb(0,2)
11390    Ret_code=8
11395 SUBEND
11400 !
11405 SUB Stir(INTEGER Data)
11410 !*************************************************
11415 !*
11420 !*    Stirer ON/OFF
11425 !*
11430 !*************************************************
11435 Stir:!
11440    SELECT Data
11445    CASE 0
11450       CALL Gmax_rb(0,1)
11455    CASE ELSE
11460       CALL Gmax_sb(0,1)
11465    END SELECT
11470 SUBEND
11475 !
11480 SUB Cover(INTEGER Mode,Ret_code)
11485 !*************************************************
11490 !*
11495 !*    Open/Close Cover
11500 !*
11505 !*************************************************
11510    OPTION BASE 1
11515    INTEGER Senser
11520    !
11525 Cover:!
11530    ON TIME (TIMEDATE+20.0) MOD 86400 RECOVER Timeout
11535    SELECT Mode
11540    CASE =0
11545       CALL Gmax_sb(0,4)
11550       LOOP
11555          CALL Gmax_di(1,5,Senser)
11560       EXIT IF Senser>0
11565       END LOOP
11570       CALL Gmax_rb(0,4)
11575    CASE =1
11580       CALL Gmax_sb(0,3)
11585       LOOP
11590          CALL Gmax_di(1,4,Senser)
11595       EXIT IF Senser>0
11600       END LOOP
11605       CALL Gmax_rb(0,3)
11610    CASE =2
11615       CALL Gmax_sb(0,4)
11620       CALL Gmax_rb(0,4)
11625    CASE =3
11630       CALL Gmax_sb(0,3)
11635       CALL Gmax_rb(0,3)
11640    END SELECT
11645    OFF TIME
11650    Ret_code=0
11655    SUBEXIT
11660 Timeout:!
11665    OFF TIME
11670    CALL Gmax_rb(0,3)
11675    CALL Gmax_rb(0,4)
11680    Ret_code=8
11685 SUBEND
```

```
11690 !
11695 SUB Tenbin(INTEGER Mode,REAL Data,INTEGER Ret_code)
11700 !*****************************************************
11705 !*
11710 !*     Get Data from Tenbin
11715 !*
11720 !*****************************************************
11725    OPTION BASE 1
11730    COM /Gmax_io/ @Gmax_io
11735    DIM Buf$[20]
11740    INTEGER I,Mini,Maxi,Retry
11745    REAL Max,Min,Datax(50)
11750    REAL X1,X2,Mean,Sigma,Cv,Kentei
11755    !
11760 Tenbin:!
11765    Retry=0
11770    ON TIMEOUT SC(@Gmax_io),2.0 GOTO Timeout
11775    SELECT Mode
11780    CASE 0
11785       OUTPUT @Gmax_io;"TA"
11790       WAIT 5.0
11795       Ret_code=0
11800    CASE ELSE
11805       ON TIME (TIMEDATE+500.0) MOD 86400 RECOVER Timeout
11810 Calculate:!
11815       FOR I=1 TO 1
11820          OUTPUT @Gmax_io;"BD"
11825          ENTER @Gmax_io;Buf$
11830          IF Buf$[1;2]<>"S " THEN GOTO Calculate
11835          X1=VAL(Buf$[5;8])
11840          DISP RPT$(" ",20)&VAL$(I)&"= "&VAL$(X1)&"       "
11845          WAIT .5
11850          Datax(I)=X1
11855       NEXT I
11860       FOR I=1 TO 1
11865          Datax(I)=Datax(I)
11870       NEXT I
11875       !
11880       DISP RPT$(" ",80)
11885       Data=Datax(1)
11890       Ret_code=0
11895    END SELECT
11900    OFF TIME
11905    OFF TIMEOUT
11910    SUBEXIT
11915 Timeout:!
11920    OFF TIME
11925    OFF TIMEOUT
11930    Ret_code=8
11935 SUBEND
11940 !
11945 SUB Titrater(INTEGER Mode,REAL Data,INTEGER Ret_code)
11950 !*****************************************************
11955 !*
11960 !*     Titrater
11965 !*
11970 !*****************************************************
11975    OPTION BASE 1
11980    COM /Rs_232c/ @Rs_232c
11985    !
11990 Titrater:!
11995    ON TIMEOUT SC(@Rs_232c),2 GOTO Timeout
12000    SELECT Mode
12005    CASE =0
12010       OUTPUT @Rs_232c;"L"
12015       OFF TIMEOUT
12020    CASE ELSE
12025       OUTPUT @Rs_232c;VAL$(Data/20)
12030       OUTPUT @Rs_232c;"M"
12035       OFF TIMEOUT
12040       WAIT 43*(Data/20)+4*(INT(Data/20)+1)
12045    END SELECT
```

```
12050    Ret_code=0
12055    SUBEXIT
12060    !
12065 Timeout:!
12070    OFF TIMEOUT
12075    Ret_code=8
12080 SUBEND
12085 !
12090 SUB Move_sampler(INTEGER Mode,REAL Data,INTEGER Ret_code)
12095 !*********************************************************
12100 !*
12105 !*     Sampler
12110 !*
12115 !*********************************************************
12120    OPTION BASE 1
12125    REAL Pulse
12130    !
12135 Move_sampler:!
12140    SELECT Mode
12145    CASE 0
12150       IF Data>0 THEN
12155          Pulse=-INT(Data*1800)
12160          CALL Motor_5(1,Pulse,Ret_code)
12165       ELSE
12170          CALL Motor_5(3,-1,Ret_code)
12175       END IF
12180    CASE 1
12185       Pulse=INT(Data*1800)
12190       CALL Motor_5(1,Pulse,Ret_code)
12195    CASE 2
12200       Pulse=1000
12205       CALL Motor_5(1,Pulse,Ret_code)
12210    END SELECT
12215 SUBEND
12220 !
12225 SUB Move_arm(INTEGER Position,Cup,Ret_code)
12230 !*************************************************
12235 !*
12240 !*    Move the arm
12245 !*
12250 !*************************************************
12255    OPTION BASE 1
12260    COM /Robot/ INTEGER Now_posi,Hand,REAL Length,Hight,Turn
12265    INTEGER I
12270    REAL L1,H1,T1,Pulse
12275    REAL Hosei_h,Hosei_l,Hosei_t
12280    REAL Chip_set,Chip_reset,Chip_ret
12285    !
12290 Move_arm:!
12295    Chip_ret=230
12300    Chip_set=-1650
12305    Chip_reset=1500
12310    SELECT Position
12315    CASE =0
12320       CALL Motor_3(4,-1,Ret_code)
12325       IF Ret_code<>0 THEN GOTO Move_err
12330       Length=0
12335       CALL Motor_1(4,-1,Ret_code)
12340       IF Ret_code<>0 THEN GOTO Move_err
12345       Hight=0
12350       CALL Motor_2(3,-1,Ret_code)
12355       IF Ret_code<>0 THEN GOTO Move_err
12360       Turn=0
12365       Now_posi=0
12370    CASE =1
12375       SELECT Cup
12380          CASE 3,4
12385             L1=0
12390          CASE 2
12395             L1=95
12400          CASE 6
12405             L1=180    ! Titrator
```

```
12410          CASE 7
12415             L1=70
12420          CASE 7,8
12425             L1=100
12430          CASE ELSE
12435             L1=0
12440       END SELECT
12445       IF Cup<>8 OR Cup<>7 THEN
12450          SELECT Hand
12455             CASE 5
12460                L1=348
12465          END SELECT
12470       END IF
12475       IF (L1-Length)<>0 THEN
12480          CALL Motor_3(2,L1-Length,Ret_code)
12485          IF Ret_code<>0 THEN GOTO Move_err
12490          Length=L1
12495       END IF
12500       Now_posi=1
12505    CASE =2
12510       SELECT Cup
12515       CASE =2
12520          H1=50
12525       CASE =3
12530          H1=12000
12535       CASE =4
12540          H1=11800
12545       CASE =6
12550          H1=6700      ! Titrator(Cup Table Ue)
12555       CASE =7
12560          H1=6700
12565       CASE =9
12570          H1=12600     ! SAMPLE no suikomi
12575       CASE =17
12580          H1=12200     ! B_Cup Tenbin Ue
12585       CASE =18
12590          H1=14000     ! B_Cup Shita Sample
12595       END SELECT
12600       IF (H1-Hight)<>0 THEN
12605          Pulse=H1-Hight
12610          CALL Motor_1(2,Pulse,Ret_code)
12615          IF Ret_code<>0 THEN GOTO Move_err
12620          Hight=Hight+Pulse
12625       END IF
12630       Now_posi=2
12635    CASE =13
12640       RESTORE Posi_data
12645       FOR I=0 TO Position
12650          READ L1,H1,T1
12655       NEXT I
12660       L2=L1-60
12665       IF (L2-Length)<>0 THEN
12670          CALL Motor_3(2,L2-Length,Ret_code)
12675          IF Ret_code<>0 THEN GOTO Move_err
12680          Length=L2
12685       END IF
12690       CALL Motor_4(1,150,Ret_code)
12695       IF Ret_code<>0 THEN GOTO Move_err
12700       IF (T1-Turn)<>0 THEN
12705          CALL Motor_2(1,T1-Turn,Ret_code)
12710          IF Ret_code<>0 THEN GOTO Move_err
12715          Turn=T1
12720       END IF
12725       IF (H1-Hight)<>0 THEN
12730          Pulse=H1-Hight
12735          CALL Motor_1(2,Pulse,Ret_code)
12740          IF Ret_code<>0 THEN GOTO Move_err
12745          Hight=Hight+Pulse
12750       END IF
12755       IF (L1-Length)<>0 THEN
12760          CALL Motor_3(2,L1-Length,Ret_code)
12765          IF Ret_code<>0 THEN GOTO Move_err
```

```
12770          Length=L1
12775       END IF
12780       Pulse=Chip_set
12785       CALL Motor_1(2,Pulse,Ret_code)
12790       IF Ret_code<>0 THEN GOTO Move_err
12795       Hight=Hight+Pulse
12800       Pulse=12600
12805       CALL Motor_1(2,Pulse,Ret_code)
12810       IF Ret_code<>0 THEN GOTO Move_err
12815       Hight=Hight+Pulse
12820       Now_posi=Position
12825    CASE =14
12830       IF Now_posi=6 THEN
12835          Pulse=Chip_reset
12840          CALL Motor_1(2,Pulse,Ret_code)
12845          IF Ret_code<>0 THEN GOTO Move_err
12850          Hight=Hight+Pulse
12855          Pulse=-Chip_reset
12860          CALL Motor_1(2,Pulse,Ret_code)
12865          IF Ret_code<>0 THEN GOTO Move_err
12870          Hight=Hight+Pulse
12875          Now_posi=6
12880       END IF
12885    CASE ELSE
12890       IF Hand=0 THEN
12895          J=Cup
12900       ELSE
12905          J=Hand
12910       END IF
12915       RESTORE Hosei
12920       FOR I=0 TO J
12925          READ Hosei_l,Hosei_h,Hosei_t
12930       NEXT I
12935       RESTORE Posi_data
12940       FOR I=0 TO Position
12945          READ L1,H1,T1
12950       NEXT I
12955    !
12960          SELECT Now_posi
12965          CASE 10,11,12
12970             Pulse=Hosei_h-Hight
12975             CALL Motor_1(2,Pulse,Ret_code)
12980             IF Ret_code<>0 THEN GOTO Move_err
12985             Hight=Hight+Pulse
12990          CASE 8
12995             IF ((Hosei_h-Hight)<>0 AND Position=15) AND Hand<>0 THEN
13000                Pulse=Hosei_h-Hight
13005                CALL Motor_1(2,Pulse,Ret_code)
13010                IF Ret_code<>0 THEN GOTO Move_err
13015                Hight=Hight+Pulse
13020             END IF
13025          CASE 7
13030             IF ((Hosei_h-Hight)<>0 AND Position=16) AND Hand<>0 THEN
13035                Pulse=Hosei_h-Hight
13040                CALL Motor_1(2,Pulse,Ret_code)
13045                IF Ret_code<>0 THEN GOTO Move_err
13050                Hight=Hight+Pulse
13055             END IF
13060          CASE 1
13065             SELECT Position
13070             CASE 7,8
13075                IF (H1-Hight)<>0 THEN
13080                   Pulse=H1-Hight
13085                   CALL Motor_1(2,Pulse,Ret_code)
13090                   Hight=Hight+Pulse
13095                END IF
13100             END SELECT
13105          END SELECT
13110          SELECT Position
13115          CASE 4,5,21,22
13120             IF (H1-Hight)<>0 THEN
13125                Pulse=H1-Hight
```

```
13130                CALL Motor_1(2,Pulse,Ret_code)
13135                Hight=Hight+Pulse
13140             END IF
13145          END SELECT
13150          SELECT Position
13155          CASE 17
13160             IF (H1-Hight)<>0 AND Hand>0 THEN
13165                Pulse=H1-Hight
13170                CALL Motor_1(2,Pulse,Ret_code)
13175                IF Ret_code<>0 THEN GOTO Move_err
13180                Hight=Hight+Pulse
13185             END IF
13190          CASE 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,21,22
13195             SELECT Now_posi
13200             CASE 0,1,2,3,6,9,10,11,12,13,14,15,16
13205                IF (T1-Turn)<>0 THEN
13210                   CALL Motor_2(1,T1-Turn,Ret_code)
13215                   IF Ret_code<>0 THEN GOTO Move_err
13220                   Turn=T1
13225                END IF
13230             CASE 4,5
13235                IF (T1-Turn)<>0 THEN
13240                   CALL Motor_2(2,T1-Turn,Ret_code)
13245                   IF Ret_code<>0 THEN GOTO Move_err
13250                   Turn=T1
13255                END IF
13260             END SELECT
13265          END SELECT
13270          IF (L1-Length)<>0 AND Now_posi=17 THEN
13275             IF (L1-Length)<>0 AND Position=18 THEN
13280                CALL Motor_3(2,L1-Length,Ret_code)
13285                IF Ret_code<>0 THEN GOTO Move_err
13290                Length=L1
13295             END IF
13300          END IF
13305          SELECT Position
13310          CASE 0,1,2,3,6,9,10,11,12,13,14,19,20
13315             SELECT Now_posi
13320             CASE 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,17,18
13325                SELECT Position
13330                   CASE 6,9
13335                      IF Hand>0 THEN
13340                         H1=H1+Chip_ret
13345                      END IF
13350                END SELECT
13355                IF (H1-Hight)<>0 THEN
13360                   Pulse=H1-Hight
13365                   CALL Motor_1(2,Pulse,Ret_code)
13370                   IF Ret_code<>0 THEN GOTO Move_err
13375                   Hight=Hight+Pulse
13380                END IF
13385             END SELECT
13390          CASE 16
13395             IF (L1-Length)<>0 AND Now_posi=7 THEN
13400                CALL Motor_3(2,L1-Length,Ret_code)
13405                IF Ret_code<>0 THEN GOTO Move_err
13410                Length=L1
13415             END IF
13420          END SELECT
13425          SELECT Now_posi
13430          CASE 7
13435             IF (Hosei_l-Length)<>0 AND Hand=0 THEN
13440                CALL Motor_3(2,Hosei_l-Length,Ret_code)
13445                IF Ret_code<>0 THEN GOTO Move_err
13450                Length=Hosei_l
13455             END IF
13460          END SELECT
13465          IF (L1-Length)<>0 AND Now_posi<>7 THEN
13470             IF (L1-Length)<>0 AND Position<>17 THEN
13475                IF (L1-Length)<>0 AND Position<>18 THEN
13480                   IF (L1-Length)<>0 AND Position<>19 THEN
13485                      IF (L1-Length)<>0 AND Position<>20 THEN
```

```
13490                          CALL Motor_3(2,L1-Length,Ret_code)
13495                          IF Ret_code<>0 THEN GOTO Move_err
13500                          Length=L1
13505                       END IF
13510                    END IF
13515                 END IF
13520              END IF
13525           END IF
13530           SELECT Now_posi
13535           CASE 4,5
13540              IF (T1-Turn)<>0 THEN
13545                 CALL Motor_2(2,T1-Turn,Ret_code)
13550                 IF Ret_code<>0 THEN GOTO Move_err
13555                 Turn=T1
13560              END IF
13565           CASE ELSE
13570              IF (T1-Turn)<>0 THEN
13575                 CALL Motor_2(1,T1-Turn,Ret_code)
13580                 IF Ret_code<>0 THEN GOTO Move_err
13585                 Turn=T1
13590              END IF
13595           END SELECT
13600           IF Hand>0 THEN
13605              SELECT Position
13610              CASE 6,9
13615                 Pulse=-Chip_ret
13620                 CALL Motor_1(2,Pulse,Ret_code)
13625                 IF Ret_code<>0 THEN GOTO Move_err
13630                 Hight=Hight+Pulse
13635              CASE 4,5,17,18
13640                 IF (H1-Hight)<>0 THEN
13645                    Pulse=H1-Hight
13650                    CALL Motor_1(2,Pulse,Ret_code)
13655                    IF Ret_code<>0 THEN GOTO Move_err
13660                    Hight=Hight+Pulse
13665                 END IF
13670              END SELECT
13675           END IF
13680           IF (L1-Length)<>0 THEN
13685              SELECT Position
13690              CASE 17,18,19,20
13695                 CALL Motor_3(2,L1-Length,Ret_code)
13700                 IF Ret_code<>0 THEN GOTO Move_err
13705                 Length=L1
13710              END SELECT
13715           END IF
13720           SELECT Now_posi
13725           CASE 15,16
13730              IF (H1-Hight)<>0 THEN
13735                 CALL Motor_1(2,H1-Hight,Ret_code)
13740                 IF Ret_code<>0 THEN GOTO Move_err
13745                 Hight=H1
13750              END IF
13755           END SELECT
13760        Now_posi=Position
13765     END SELECT
13770 Move_err:   !
13775     SUBEXIT
13780 !
13785 Posi_data:!
13790     !      L   ,  H   ,  T
13795     DATA 0     ,0     ,0       !0   Reset Position
13800     DATA 0     ,0     ,0       !1   Ittei Length
13805     DATA 0     ,0     ,0       !2   Ittei Hight
13810     DATA 0     ,0     ,0       !3
13815     DATA 1375  ,18000,1800    !4   Sample A
13820     DATA 1275  ,18000,1915    !5   Sample B
13825     DATA 730   ,11550,490     !6   Sampling Souchi
13830     DATA 380   ,3350 ,2540    !7   Tenbin C_Cup
13835     DATA 350   ,440  ,2540    !8   Tenbin B_Cup
13840     DATA 350   ,11300,4760    !9   Titrator
13845     DATA 450   ,3600 ,3760    !10  C_Cup
```

```
13850     DATA 530   ,4350 ,1290    !11  B_Cup  Ue    (11 - 20)
13855     DATA 340   ,820  ,1290    !12  B_Cup  Shita(01 - 10)
13860     DATA 249   ,8000 ,66      !13  Chip-set
13865     DATA 730   ,13050,480     !14  Chip-reset
13870     DATA 195   ,1500 ,1850    !15  Cover Open (B_Cup)
13875     DATA 0     ,6000 ,3280    !16  Cover Open (C_Cup)
13880     DATA 330   ,16500,2555    !17  Chip B_Cup Tenbin Ue
13885     DATA 0     ,0    ,0       !18
13890     DATA 440   ,20500,3765    !19  Chip C_Cup Ue
13895     DATA 380   ,20500,2540    !20  Chip C_Cup Tenbin Ue
13900     DATA 410   ,9800 ,2550    !21  Titrator C_Cup Tenbin Ue
13905     DATA 455   ,11300,3750    !22  Titrator C_Cup Ue
13910 Hosei:!
13915     DATA 0     ,0    ,0       !0   Reset Arm
13920     DATA 0     ,0    ,0       !1
13925     DATA 0     ,5500 ,0       !2   C_Cup
13930     DATA 0     ,5100 ,0       !3   B_Cup Ue    (11 - 20)
13935     DATA 0     ,1500 ,0       !4   B_Cup Shita(01 - 10)
13940     DATA -10   ,14600,12800   !5   Sampling Souchi
13945     DATA -10   ,13000,4000    !6   Titrator
13950     DATA 0     ,0    ,0       !7   Chip C_cup Ue
13955     DATA 0     ,0    ,0       !8   Chip C_cup Tenbin Ue8584
13960     DATA 0     ,0    ,0       !9   SAMPLE A,B
13965     DATA 0     ,0    ,0       !10
13970 SUBEND
13975 !
13980 SUB Move_hand(INTEGER Position,Cup,Ret_code)
13985 !*************************************************
13990 !*
13995 !*     Open/Close the hand
14000 !*
14005 !*************************************************
14010     OPTION BASE 1
14015     COM /Robot/ INTEGER Now_posi,Hand,REAL Length,Hight,Turn
14020     INTEGER I
14025     REAL Pulse
14030     !
14035 Move_hand:!
14040     SELECT Position
14045     CASE =0
14050        IF Cup=2 THEN
14055           Pulse=-400
14060           CALL Motor_4(1,Pulse,Ret_code)
14065        END IF
14070           CALL Motor_4(3,-1,Ret_code)
14075           Hand=0
14080     CASE =1
14085        RESTORE Posi_data
14090        FOR I=1 TO Cup
14095           READ Pulse
14100        NEXT I
14105        CALL Motor_4(1,Pulse,Ret_code)
14110        Hand=Cup
14115        Pulse=50
14120        CALL Motor_4(1,Pulse,Ret_code)
14125     CASE 2
14130        Pulse=140
14135        CALL Motor_4(1,Pulse,Ret_code)
14140        Hand=0
14145     CASE 3
14150        Pulse=150
14155        CALL Motor_4(1,Pulse,Ret_code)
14160     END SELECT
14165     SUBEXIT
14170 !
14175 !------Position Data
14180 !
14185 Posi_data:!
14190     DATA 900,900,900,900,900,900,900,900,900,900,900
14195 SUBEND
14200 !
14205 SUB Motor_1(INTEGER Mode,REAL Pulse,INTEGER Ret_code)
```

```
14210 !*******************************************************
14215 !*
14220 !*     Handling pulse-motor (hight)
14225 !*
14230 !*******************************************************
14235    OPTION BASE 1
14240    COM /Gmax_pm/ @Gmax_pm
14245    COM /Robot/ INTEGER Now_posi,Hand,REAL Length,Hight,Turn
14250    DIM P1$[2],P2$[7],Buf$[32]
14255    INTEGER I
14260    REAL Temp
14265    !
14270 Motor_1:!
14275    IF Pulse<0 THEN
14280       P1$="0"
14285       P2$=VAL$(-Pulse)
14290    ELSE
14295       P1$="1"
14300       P2$=VAL$(Pulse)
14305    END IF
14310    ON TIMEOUT SC(@Gmax_pm),5.0 GOTO Timeout
14315    ON INTR SC(@Gmax_pm) GOTO Timeout
14320    ENABLE INTR SC(@Gmax_pm);2
14325    !
14330    IF Mode>0 AND Mode<8 THEN
14335       I=32
14340       WHILE BIT(I,5)=1
14345          OUTPUT @Gmax_pm;"DS,1,2"
14350          ENTER @Gmax_pm;I
14355       END WHILE
14360    END IF
14365    SELECT Mode
14370    CASE 0
14375       Buf$="MI,1,0,2, 50,350,500 "
14380    CASE 1
14385       Buf$="MC,1,"&P1$&","&P2$&",350"
14390    CASE 2
14395       Buf$="MA,1,"&P1$&","&P2$
14400    CASE 3
14405       Buf$="ML,1,"&P1$&",350"
14410    CASE 4
14415       Buf$="MH,1,"&P1$
14420    CASE 5
14425       Buf$="MG,1,"&P1$&",350"
14430    CASE 6
14435       Buf$="MP,1,"&P1$
14440    CASE 7
14445       Buf$="MS,1,"&P1$
14450    CASE 8
14455       Buf$="DS,1,"&P2$
14460    END SELECT
14465    OUTPUT @Gmax_pm;Buf$
14470    IF Mode>0 AND Mode<8 THEN
14475       I=32
14480       WHILE BIT(I,5)=1
14485          OUTPUT @Gmax_pm;"DS,1,2"
14490          ENTER @Gmax_pm;I
14495       END WHILE
14500       OUTPUT @Gmax_pm;"DS,1,4"
14505       ENTER @Gmax_pm;Temp
14510       IF Temp<>0 THEN
14515          IF Pulse>0 THEN
14520             Pulse=Pulse-(Temp+1)
14525          ELSE
14530             Pulse=Pulse+(Temp+1)
14535          END IF
14540       END IF
14545    END IF
14550    IF Mode=8 THEN ENTER @Gmax_pm;Pulse
14555    OFF TIMEOUT
14560    OFF INTR
14565    Ret_code=0
```

```
14570    SUBEXIT
14575 Timeout:!
14580    OFF TIMEOUT
14585    OFF INTR
14590    Ret_code=8
14595 SUBEND
14600 !
14605 SUB Motor_2(INTEGER Mode,REAL Pulse,INTEGER Ret_code)
14610 !*****************************************************
14615 !*
14620 !*      for arm (rotate)
14625 !*
14630 !*****************************************************
14635    OPTION BASE 1
14640    COM /Gmax_pm/ @Gmax_pm
14645    COM /Robot/ INTEGER Now_posi,Hand,REAL Length,Hight,Turn
14650    DIM P1$[2],P2$[7],Buf$[20]
14655    INTEGER I
14660    !
14665 Motor_2:!
14670    IF Pulse<0 THEN
14675       P1$="1"
14680       P2$=VAL$(-Pulse)
14685    ELSE
14690       P1$="0"
14695       P2$=VAL$(Pulse)
14700    END IF
14705    ON TIMEOUT SC(@Gmax_pm),5.0 GOTO Timeout
14710    ON INTR SC(@Gmax_pm) GOTO Timeout
14715    ENABLE INTR SC(@Gmax_pm);2
14720    !
14725    IF Mode>0 AND Mode<8 THEN
14730       I=32
14735       WHILE BIT(I,5)=1
14740          OUTPUT @Gmax_pm;"DS,0,2"
14745          ENTER @Gmax_pm;I
14750       END WHILE
14755    END IF
14760    SELECT Mode
14765    CASE 0
14770       Buf$="MI,0,0,2,50,600,1000"
14775    CASE 1
14780       Buf$="MC,0,"&P1$&","&P2$&",600"
14785    CASE 2
14790       Buf$="MA,0,"&P1$&","&P2$
14795    CASE 3
14800       Buf$="ML,0,"&P1$&",600"
14805    CASE 4
14810       Buf$="MH,0,"&P1$
14815    CASE 5
14820       Buf$="MG,0,"&P1$&",600"
14825    CASE 6
14830       Buf$="MP,0,"&P1$
14835    CASE 7
14840       Buf$="MS,0,"&P1$
14845    CASE 8
14850       Buf$="DS,0,"&P2$
14855    END SELECT
14860    OUTPUT @Gmax_pm;Buf$
14865    IF Mode>0 AND Mode<8 THEN
14870       I=32
14875       WHILE BIT(I,5)=1
14880          OUTPUT @Gmax_pm;"DS,0,2"
14885          ENTER @Gmax_pm;I
14890       END WHILE
14895    END IF
14900    IF Mode=8 THEN ENTER @Gmax_pm;Pulse
14905    OFF TIMEOUT
14910    OFF INTR
14915    Ret_code=0
14920    SUBEXIT
14925 Timeout:!
```

```
14930    OFF TIMEOUT
14935    OFF INTR
14940    Ret_code=8
14945 SUBEND
14950 !
14955 SUB Motor_3(INTEGER Mode,REAL Pulse,INTEGER Ret_code)
14960 !*****************************************************
14965 !*
14970 !*       for arm (stretch)
14975 !*
14980 !*****************************************************
14985    OPTION BASE 1
14990    COM /Gmax_pm/ @Gmax_pm
14995    COM /Robot/ INTEGER Now_posi,Hand,REAL Length,Hight,Turn
15000    DIM P1$[2],P2$[7],Buf$[20]
15005    INTEGER I
15010    !
15015 Motor_3:!
15020    IF Pulse<0 THEN
15025       P1$="0"
15030       P2$=VAL$(-Pulse)
15035    ELSE
15040       P1$="1"
15045       P2$=VAL$(Pulse)
15050    END IF
15055    ON TIMEOUT SC(@Gmax_pm),5.0 GOTO Timeout
15060    ON INTR SC(@Gmax_pm) GOTO Timeout
15065    ENABLE INTR SC(@Gmax_pm);2
15070    !
15075    IF Mode>0 AND Mode<8 THEN
15080       I=32
15085       WHILE BIT(I,5)=1
15090          OUTPUT @Gmax_pm;"DS,2,2"
15095          ENTER @Gmax_pm;I
15100       END WHILE
15105    END IF
15110    SELECT Mode
15115    CASE 0
15120       Buf$="MI,2,0,2,50,400,400"
15125    CASE 1
15130       Buf$="MC,2,"&P1$&","&P2$&",400"
15135    CASE 2
15140       Buf$="MA,2,"&P1$&","&P2$
15145    CASE 3
15150       Buf$="ML,2,"&P1$&",400"
15155    CASE 4
15160       Buf$="MH,2,"&P1$
15165    CASE 5
15170       Buf$="MG,2,"&P1$&",400"
15175    CASE 6
15180       Buf$="MP,2,"&P1$
15185    CASE 7
15190       Buf$="MS,2,"&P1$
15195    CASE 8
15200       Buf$="DS,2,"&P2$
15205    END SELECT
15210    OUTPUT @Gmax_pm;Buf$
15215    IF Mode>0 AND Mode<8 THEN
15220       I=32
15225       WHILE BIT(I,5)=1
15230          OUTPUT @Gmax_pm;"DS,2,2"
15235          ENTER @Gmax_pm;I
15240       END WHILE
15245    END IF
15250    IF Mode=8 THEN ENTER @Gmax_pm;Pulse
15255    OFF TIMEOUT
15260    OFF INTR
15265    Ret_code=0
15270    SUBEXIT
15275 Timeout:!
15280    OFF TIMEOUT
15285    OFF INTR
```

```
15290    Ret_code=8
15295 SUBEND
15300 !
15305 SUB Motor_4(INTEGER Mode,REAL Pulse,INTEGER Ret_code)
15310 !*****************************************************
15315 !*
15320 !*       for hand
15325 !*
15330 !*****************************************************
15335    OPTION BASE 1
15340    COM /Gmax_pm/ @Gmax_pm
15345    DIM P1$[2],P2$[7],Buf$[20]
15350    INTEGER I
15355    !
15360 Motor_4:!
15365    IF Pulse<0 THEN
15370       P1$="0"
15375       P2$=VAL$(-Pulse)
15380    ELSE
15385       P1$="1"
15390       P2$=VAL$(Pulse)
15395    END IF
15400    ON TIMEOUT SC(@Gmax_pm),5.0 GOTO Timeout
15405    ON INTR SC(@Gmax_pm) GOTO Timeout
15410    ENABLE INTR SC(@Gmax_pm);2
15415    !
15420    IF Mode>0 AND Mode<8 THEN
15425       I=32
15430       WHILE BIT(I,5)=1
15435          OUTPUT @Gmax_pm;"DS,3,2"
15440          ENTER @Gmax_pm;I
15445       END WHILE
15450    END IF
15455    SELECT Mode
15460    CASE 0
15465       Buf$="MI,3,0,2,50,300,500"
15470    CASE 1
15475       Buf$="MC,3,"&P1$&","&P2$&",300"
15480    CASE 2
15485       Buf$="MA,3,"&P1$&","&P2$
15490    CASE 3
15495       Buf$="ML,3,"&P1$&",300"
15500    CASE 4
15505       Buf$="MH,3,"&P1$
15510    CASE 5
15515       Buf$="MG,3,"&P1$&",300"
15520    CASE 6
15525       Buf$="MP,3,"&P1$
15530    CASE 7
15535       Buf$="MS,3,"&P1$
15540    CASE 8
15545       Buf$="DS,3,"&P2$
15550    END SELECT
15555    OUTPUT @Gmax_pm;Buf$
15560    IF Mode>0 AND Mode<8 THEN
15565       I=32
15570       WHILE BIT(I,5)=1
15575          OUTPUT @Gmax_pm;"DS,3,2"
15580          ENTER @Gmax_pm;I
15585       END WHILE
15590    END IF
15595    IF Mode=8 THEN ENTER @Gmax_pm;Pulse
15600    OFF TIMEOUT
15605    OFF INTR
15610    Ret_code=0
15615    SUBEXIT
15620 Timeout:!
15625    OFF TIMEOUT
15630    OFF INTR
15635    Ret_code=8
15640 SUBEND
15645 !
```

```
15650 SUB Motor_5(INTEGER Mode,REAL Pulse,INTEGER Ret_code)
15655 !*****************************************************
15660 !*
15665 !*     for sampling
15670 !*
15675 !*****************************************************
15680    OPTION BASE 1
15685    COM /Gmax_pm/ @Gmax_pm
15690    DIM P1$[2],P2$[7],Buf$[20]
15695    INTEGER I
15700    !
15705 Motor_5:!
15710    IF Pulse<0 THEN
15715       P1$="1"
15720       P2$=VAL$(-Pulse)
15725    ELSE
15730       P1$="0"
15735       P2$=VAL$(Pulse)
15740    END IF
15745    ON TIMEOUT SC(@Gmax_pm),5.0 GOTO Timeout
15750    ON INTR SC(@Gmax_pm) GOTO Timeout
15755    ENABLE INTR SC(@Gmax_pm);2
15760    !
15765    IF Mode>0 AND Mode<8 THEN
15770       I=32
15775       WHILE BIT(I,5)=1
15780          OUTPUT @Gmax_pm;"DS,4,2"
15785          ENTER @Gmax_pm;I
15790       END WHILE
15795    END IF
15800    SELECT Mode
15805    CASE 0
15810       Buf$="MI,4,0,2,50,300,500"
15815    CASE 1
15820       Buf$="MC,4,"&P1$&","&P2$&",300"
15825    CASE 2
15830       Buf$="MA,4,"&P1$&","&P2$
15835    CASE 3
15840       Buf$="ML,4,"&P1$&",400"
15845    CASE 4
15850       Buf$="MH,4,"&P1$
15855    CASE 5
15860       Buf$="MG,4,"&P1$&",300"
15865    CASE 6
15870       Buf$="MP,4,"&P1$
15875    CASE 7
15880       Buf$="MS,4,"&P1$
15885    CASE 8
15890       Buf$="DS,4,"&P2$
15895    END SELECT
15900    OUTPUT @Gmax_pm;Buf$
15905    IF Mode>0 AND Mode<8 THEN
15910       I=32
15915       WHILE BIT(I,5)=1
15920          OUTPUT @Gmax_pm;"DS,4,2"
15925          ENTER @Gmax_pm;I
15930       END WHILE
15935    END IF
15940    IF Mode=8 THEN ENTER @Gmax_pm;Pulse
15945    OFF TIMEOUT
15950    OFF INTR
15955    Ret_code=0
15960    SUBEXIT
15965 Timeout:!
15970    OFF TIMEOUT
15975    OFF INTR
15980    Ret_code=8
15985 SUBEND
15990 !
15995 SUB Gmax_di(INTEGER Ch_no,Bit_no,Data)
16000 !*************************************
16005 !*
```

```
16010 !*     Digital Input from GMACS
16015 !*
16020 !****************************************
16025    OPTION BASE 1
16030    COM /Gmax_io/ @Gmax_io
16035    !
16040 Gmax_di:!
16045    ON TIMEOUT SC(@Gmax_io),2.0 GOTO Timeout
16050    ON INTR SC(@Gmax_io) GOTO Timeout
16055    ENABLE INTR SC(@Gmax_io);2
16060    OUTPUT @Gmax_io;"DI,"&VAL$(Ch_no)
16065    ENTER @Gmax_io;Data
16070    IF Bit_no>=0 THEN
16075       Data=BINAND(Data,INT(2^Bit_no))
16080    END IF
16085 Timeout:!
16090    OFF TIMEOUT
16095    OFF INTR
16100 SUBEND
16105 !
16110 SUB Gmax_dr
16115 !****************************************
16120 !*
16125 !*    Digital Output to GMACS
16130 !*
16135 !****************************************
16140    OPTION BASE 1
16145    COM /Gmax_io/ @Gmax_io
16150    COM /Gmax_pm/ @Gmax_pm
16155    !
16160 Gmax_dr:!
16165    ON TIMEOUT SC(@Gmax_io),2.0 GOTO Timeout
16170    OUTPUT @Gmax_io;"DR"
16175    OFF TIMEOUT
16180    ON TIMEOUT SC(@Gmax_pm),2.0 GOTO Timeout
16185    OUTPUT @Gmax_pm;"DR"
16190 Timeout:!
16195    OFF TIMEOUT
16200 SUBEND
16205 !
16210 SUB Gmax_sb(INTEGER Ch_no,Bit_no)
16215 !****************************************
16220 !*
16225 !*    Digital Output to GMACS
16230 !*
16235 !****************************************
16240    OPTION BASE 1
16245    COM /Gmax_io/ @Gmax_io
16250    INTEGER Data
16255    !
16260 Gmax_sb:!
16265    Data=2^Bit_no
16270    ON TIMEOUT SC(@Gmax_io),2.0 GOTO Timeout
16275    ON INTR SC(@Gmax_io) GOTO Timeout
16280    ENABLE INTR SC(@Gmax_io);2
16285    OUTPUT @Gmax_io;"SB,"&VAL$(Ch_no)&","&VAL$(Data)
16290 Timeout:!
16295    OFF TIMEOUT
16300    OFF INTR
16305 SUBEND
16310 !
16315 SUB Gmax_rb(INTEGER Ch_no,Bit_no)
16320 !****************************************
16325 !*
16330 !*    Hold Digital Output to GMACS
16335 !*
16340 !****************************************
16345    OPTION BASE 1
16350    COM /Gmax_io/ @Gmax_io
16355    INTEGER Data
16360    !
16365 Gmax_rb:!
```

```
16370    Data=2^Bit_no
16375    ON TIMEOUT SC(@Gmax_io),2.0 GOTO Timeout
16380    ON INTR SC(@Gmax_io) GOTO Timeout
16385    ENABLE INTR SC(@Gmax_io);2
16390    OUTPUT @Gmax_io;"RB,"&VAL$(Ch_no)&","&VAL$(Data)
16395 Timeout:!
16400    OFF TIMEOUT
16405    OFF INTR
16410 SUBEND
16415 !
16420 SUB Disp_measure(INTEGER State)
16425 !*************************************
16430 !*
16435 !*      Display
16440 !*
16445 !*************************************
16450    OPTION BASE 1
16455    COM /Result/ REAL Sample_1st(*),Dilution(*)
16460    COM /Result/ REAL Sample_2nd(*),INTEGER Cup_no(*),REAL Mea_vial(*)
16465    COM /Measure/ INTEGER Solution,Sampling,Max_cup,Max_bottle
16470    COM /Measure/ INTEGER Now_cup,Now_bottle,S2_count
16475    DIM Buf$[50]
16480    INTEGER X,Y,I
16485    !
16490 Disp_measure:!
16495    SELECT State
16500    CASE 0
16505       GCLEAR
16510       OUTPUT 2;"K";
16515       SELECT Sampling
16520       CASE 1
16525          IF Solution=1 THEN
16530             Buf$="  1st Sampling  (A)  "
16535          ELSE
16540             Buf$="  1st Sampling  (B)  "
16545          END IF
16550       CASE 2
16555          IF Solution=1 THEN
16560             Buf$="  Dilution      (A)  "
16565          ELSE
16570             Buf$="  Dilution      (B)  "
16575          END IF
16580       CASE 3
16585          IF Solution=1 THEN
16590             Buf$="  2nd Sampling  (A)  "
16595          ELSE
16600             Buf$="  2nd Sampling  (B)  "
16605          END IF
16610       END SELECT
16615       PRINT TABXY(15,1);Buf$
16620       !
16625       SELECT Sampling
16630       CASE 1,2
16635          MOVE 287,194
16640          GOSUB Frame_1
16645          PRINT TABXY(59,10);" No.  Data (g) "
16650          FOR I=1 TO Max_cup
16655             PRINT TABXY(59,I+10);"  "&VAL$(I)
16660             PRINT TABXY(53+I*5,5);VAL$(I)
16665             CALL Disp_b(3,287+25*(I-1),210)
16670          NEXT I
16675          CALL Disp_b(5,120,195)
16680          CALL Disp_b(6,120,85)
16685       CASE 3
16690          MOVE 232,218
16695          GOSUB Frame_2
16700          MOVE 317,218
16705          GOSUB Frame_2
16710          PRINT TABXY(48,8);" No.  Data (g) "
16715          PRINT TABXY(65,8);" No.  Data (g) "
16720          FOR I=1 TO Max_bottle
16725             X=47+((I-1) DIV 10)*17
```

```
16730             Y=((I-1) MOD 10)+9
16735             PRINT TABXY(X,Y);RPT$(" ",4-LEN(VAL$(I)))&VAL$(I)
16740             X=49+3*((I-1) MOD 10)
16745             IF Max_bottle>10 THEN
16750                Y=1+((I-1) DIV 10)*3
16755             ELSE
16760                Y=4
16765             END IF
16770             PRINT TABXY(X,Y);VAL$(I)
16775             X=243+15*((I-1) MOD 10)
16780             IF Max_bottle>10 THEN
16785                Y=270-((I-1) DIV 10)*36
16790             ELSE
16795                Y=234
16800             END IF
16805             CALL Disp_b(1,X,Y)
16810          NEXT I
16815          CALL Disp_b(5,120,195)
16820          CALL Disp_b(6,120,85)
16825       END SELECT
16830       !
16835    CASE 1
16840       PEN 1
16845       GOSUB Line_1
16850    CASE 2
16855       PEN 1
16860       GOSUB Line_2
16865    CASE 3
16870       PEN 1
16875       GOSUB Line_3
16880    CASE 4
16885       PEN 1
16890       GOSUB Line_4
16895    CASE 5
16900       PEN 1
16905       GOSUB Line_5
16910    CASE 6
16915       PEN 1
16920       GOSUB Line_6
16925    CASE 7
16930       PEN 1
16935       GOSUB Line_7
16940    CASE 8
16945       PEN 1
16950       GOSUB Line_8
16955    CASE 9
16960       PEN 1
16965       GOSUB Line_9
16970    CASE 10
16975       PEN 1
16980       GOSUB Line_10
16985    CASE 11
16990       PEN 1
16995       GOSUB Line_11
17000    CASE 12
17005       PEN 1
17010       GOSUB Line_12
17015    CASE 13
17020       PEN -1
17025       GOSUB Line_1
17030       GOSUB Line_2
17035       GOSUB Line_3
17040       GOSUB Line_4
17045       GOSUB Line_5
17050       GOSUB Line_6
17055       GOSUB Line_7
17060       GOSUB Line_8
17065       GOSUB Line_9
17070       GOSUB Line_10
17075       GOSUB Line_11
17080       GOSUB Line_12
17085       PEN 1
```

```
17090     CASE 14
17095        CALL Disp_b(7,185,195)
17100     CASE 15
17105        SELECT Sampling
17110        CASE 1
17115           FOR I=1 TO Now_cup
17120              IF Sample_1st(Solution,I)>0 THEN
17125                 Temp=INT(Sample_1st(Solution,I)*10000+.5)/10000
17130                 PRINT TABXY(65,I+10);
17135                 PRINT VAL$(Temp)&RPT$(" ",9-LEN(VAL$(Temp)))
17140                 PRINT TABXY(53+I*5,5);VAL$(I)
17145                 CALL Disp_b(10,287+25*(I-1),210)
17150              END IF
17155           NEXT I
17160        CASE 2
17165           FOR I=1 TO Now_cup
17170              IF Dilution(Solution,I)>0 THEN
17175                 Temp=INT(Dilution(Solution,I)*10000+.5)/10000
17180                 PRINT TABXY(65,I+10);
17185                 PRINT VAL$(Temp)&RPT$(" ",9-LEN(VAL$(Temp)))
17190                 PRINT TABXY(53+I*5,5);VAL$(I)
17195                 CALL Disp_b(4,287+25*(I-1),210)
17200              END IF
17205           NEXT I
17210        CASE 3
17215           FOR I=1 TO Now_bottle
17220              IF Sample_2nd(Solution,I)>0 THEN
17225                 Temp=INT(Sample_2nd(Solution,I)*10000+.5)/10000
17230                 X=54+((I-1) DIV 10)*17
17235                 Y=((I-1) MOD 10)+9
17240                 PRINT TABXY(X,Y);VAL$(Temp)&RPT$(" ",9-LEN(VAL$(Temp)))
17245                 X=49+3*((I-1) MOD 10)
17250                 IF Max_bottle>10 THEN
17255                    Y=1+((I-1) DIV 10)*3
17260                 ELSE
17265                    Y=4
17270                 END IF
17275                 PRINT TABXY(X,Y);VAL$(I)
17280                 X=243+15*((I-1) MOD 10)
17285                 IF Max_bottle>10 THEN
17290                    Y=270-((I-1) DIV 10)*36
17295                 ELSE
17300                    Y=234
17305                 END IF
17310                 CALL Disp_b(2,X,Y)
17315              END IF
17320           NEXT I
17325        END SELECT
17330     CASE 16
17335        CALL Disp_b(1,120,119)
17340     CASE 17
17345        PEN -1
17350        CALL Disp_b(1,120,119)
17355        PEN 1
17360     CASE 18
17365        CALL Disp_b(2,120,119)
17370     CASE 19
17375        PEN -1
17380        CALL Disp_b(2,120,119)
17385        PEN 1
17390     CASE 20
17395        CALL Disp_b(3,120,119)
17400     CASE 21
17405        PEN -1
17410        CALL Disp_b(3,120,119)
17415        PEN 1
17420     CASE 22
17425        CALL Disp_b(4,120,119)
17430     CASE 23
17435        PEN -1
17440        CALL Disp_b(4,120,119)
17445        PEN 1
```

```
17450    CASE 24
17455       CALL Disp_b(1,54,111)
17460    CASE 25
17465       PEN -1
17470       CALL Disp_b(1,54,111)
17475       PEN 1
17480    CASE 26
17485       CALL Disp_b(2,54,111)
17490    CASE 27
17495       PEN -1
17500       CALL Disp_b(2,54,111)
17505       PEN 1
17510    CASE 28
17515       CALL Disp_b(1,66,103)
17520    CASE 29
17525       PEN -1
17530       CALL Disp_b(1,66,103)
17535       PEN 1
17540    CASE 30
17545       CALL Disp_b(2,66,103)
17550    CASE 31
17555       PEN -1
17560       CALL Disp_b(2,66,103)
17565       PEN 1
17570    CASE 32
17575       CALL Disp_b(3,174,107)
17580    CASE 33
17585       PEN -1
17590       CALL Disp_b(3,174,107)
17595       PEN 1
17600    CASE 34
17605       CALL Disp_b(4,174,107)
17610    CASE 35
17615       PEN -1
17620       CALL Disp_b(4,174,107)
17625       PEN 1
17630    CASE 36
17635       CALL Disp_b(2,55,195)
17640    CASE 37
17645       PEN -1
17650       CALL Disp_b(2,55,195)
17655       PEN 1
17660    CASE 38
17665       CALL Disp_b(8,40,85)
17670    CASE 39
17675       PEN -1
17680       CALL Disp_b(8,40,85)
17685       PEN 1
17690    CASE 40
17695       CALL Disp_b(9,190,85)
17700    CASE 41
17705       PEN -1
17710       CALL Disp_b(9,190,85)
17715       PEN 1
17720    CASE 42
17725       CALL Disp_b(7,174,107)
17730    CASE 43
17735       PEN -1
17740       CALL Disp_b(7,174,107)
17745       PEN 1
17750    CASE 44
17755       PRINT TABXY(38,14);"Cup_no."
17760       PRINT TABXY(42,15);VAL$(Now_cup)
17765    CASE 45
17770       PRINT TABXY(3,14);"Vial_no."
17775       PRINT TABXY(7,15);VAL$(Now_bottle)
17780    CASE 46
17785       CALL Disp_b(10,120,119)
17790    CASE 47
17795       PEN -1
17800       CALL Disp_b(10,120,119)
17805       PEN 1
```

```
17810     CASE 48
17815        CALL Disp_b(10,174,107)
17820     CASE 49
17825        PEN -1
17830        CALL Disp_b(10,174,107)
17835        PEN 1
17840     END SELECT
17845     SUBEXIT
17850     !
17855 Frame_1:!
17860     RPLOT 0,0
17865     RPLOT 80,0
17870     RPLOT 80,-64
17875     RPLOT 0,-64
17880     RPLOT 0,0
17885     RPLOT 0,-14
17890     RPLOT 80,-14,2
17895     RPLOT 25,0
17900     RPLOT 25,-64
17905     RETURN
17910 Frame_2:!
17915     RPLOT 0,0
17920     RPLOT 80,0
17925     RPLOT 80,-136
17930     RPLOT 0,-136
17935     RPLOT 0,0
17940     RPLOT 0,-14
17945     RPLOT 80,-14,2
17950     RPLOT 25,0
17955     RPLOT 25,-136
17960     RETURN
17965 Line_1:! Shiryou --- Tenbin
17970     MOVE 55,192
17975     LINE TYPE 4,15
17980     RPLOT 0,0
17985     RPLOT 0,-23
17990     RPLOT 65,-23
17995     RPLOT 65,-46,2
18000     LINE TYPE 1
18005     RPLOT 65,-46
18010     RPLOT 62,-40,2
18015     RPLOT 65,-46
18020     RPLOT 68,-40,2
18025     RETURN
18030 Line_2:! Titrator --- Cup
18035     MOVE 120,192
18040     LINE TYPE 4,15
18045     RPLOT -10,0
18050     RPLOT -10,-23
18055     RPLOT 0,-23
18060     RPLOT 54,-23
18065     RPLOT 54,-56,2
18070     LINE TYPE 1
18075     RPLOT 54,-56
18080     RPLOT 51,-50,2
18085     RPLOT 54,-56
18090     RPLOT 57,-50,2
18095     RETURN
18100 Line_3:! Cup(Ue-Hyouji) --- Tenbin
18105     MOVE 120,146
18110     LINE TYPE 4,15
18115     RPLOT 0,0
18120     RPLOT 0,23
18125     RPLOT 54,23
18130     RPLOT 54,-10,2
18135     LINE TYPE 1
18140     RPLOT 0,0
18145     RPLOT -3,6,2
18150     RPLOT 0,0
18155     RPLOT 3,6,2
18160     RETURN
18165 Line_4:! Bial(Ue) === Tenbin <Idou>
```

```
18170     MOVE 120,146
18175     RPLOT 0,0
18180     RPLOT 0,23
18185     RPLOT -66,23
18190     RPLOT -66,-14,2
18195     RPLOT 0,0
18200     RPLOT -3,6,2
18205     RPLOT 0,0
18210     RPLOT 3,6,2
18215     RETURN
18220 Line_5:! Tenbin === Bial(Ue) <Idou>
18225     MOVE 120,146
18230     RPLOT 0,0
18235     RPLOT 0,23
18240     RPLOT -66,23
18245     RPLOT -66,-14,2
18250     RPLOT -66,-14
18255     RPLOT -69,-8,2
18260     RPLOT -66,-14
18265     RPLOT -63,-8
18270     RETURN
18275 Line_6:! Bial(Shita) === Tenbin <Idou>
18280     MOVE 120,146
18285     RPLOT 0,0
18290     RPLOT 0,23
18295     RPLOT -54,23
18300     RPLOT -54,-22,2
18305     RPLOT 0,0
18310     RPLOT -3,6,2
18315     RPLOT 0,0
18320     RPLOT 3,6,2
18325     RETURN
18330 Line_7:! Tenbin === Bial(Shita) <Idou>
18335     MOVE 120,146
18340     RPLOT 0,0
18345     RPLOT 0,23
18350     RPLOT -54,23
18355     RPLOT -54,-22,2
18360     RPLOT -54,-22
18365     RPLOT -57,-16,2
18370     RPLOT -54,-22
18375     RPLOT -51,-16,2
18380     RETURN
18385 Line_8:! Cup === Tenbin <Idou>
18390     MOVE 120,146
18395     RPLOT 0,0
18400     RPLOT 0,23
18405     RPLOT 54,23
18410     RPLOT 54,-10,2
18415     RPLOT 0,0
18420     RPLOT -3,6,2
18425     RPLOT 0,0
18430     RPLOT 3,6,2
18435     RETURN
18440 Line_9: ! Tenbin === Cup <Idou>
18445     MOVE 120,146
18450     RPLOT 0,0
18455     RPLOT 0,23
18460     RPLOT 54,23
18465     RPLOT 54,-10,2
18470     RPLOT 54,-10
18475     RPLOT 51,-4,2
18480     RPLOT 54,-10
18485     RPLOT 57,-4,2
18490     RETURN
18495 Line_10:! Shiryou --- Cup
18500     MOVE 55,192
18505     LINE TYPE 4,15
18510     RPLOT 0,0
18515     RPLOT 0,-23
18520     RPLOT 119,-23
18525     RPLOT 119,-56,2
```

```
18530    LINE TYPE 1
18535    RPLOT 119,-56
18540    RPLOT 116,-50.2
18545    RPLOT 119,-56
18550    RPLOT 122,-50.2
18555    RETURN
18560 Line_11:! Cup --- Bial(Ue)
18565    MOVE 174,136
18570    LINE TYPE 4,15
18575    RPLOT 0,0
18580    RPLOT 0,33
18585    RPLOT -120,33
18590    RPLOT -120,-4.2
18595    LINE TYPE 1
18600    RPLOT -120,-4
18605    RPLOT -123,2.2
18610    RPLOT -120,-4
18615    RPLOT -117,2.2
18620    RETURN
18625 Line_12:! Cup --- Bial(Shita)
18630    MOVE 174,136
18635    LINE TYPE 4,15
18640    RPLOT 0,0
18645    RPLOT 0,33
18650    RPLOT -108,33
18655    RPLOT -108,-12.2
18660    LINE TYPE 1
18665    RPLOT -108,-12
18670    RPLOT -111,-6.2
18675    RPLOT -108,-12
18680    RPLOT -105,-6.2
18685    RETURN
18690 SUBEND
18695 !
18700 SUB Disp_b(INTEGER No,X,Y)
18705 !********************
18710 !*  PLOT
18715 !********************
18720    MOVE X,Y
18725    SELECT No
18730    CASE 1
18735       GOSUB Bottle_0
18740    CASE 2
18745       GOSUB Bottle_1
18750    CASE 3
18755       GOSUB Cup_0
18760    CASE 4
18765       GOSUB Cup_1
18770    CASE 5
18775       GOSUB Titrater
18780    CASE 6
18785       GOSUB Tenbin
18790    CASE 7
18795       GOSUB Stirer
18800    CASE 8
18805       GOSUB B_table
18810    CASE 9
18815       GOSUB C_table
18820    CASE 10
18825       GOSUB Cup_2
18830    END SELECT
18835    SUBEXIT
18840    !
18845 Bottle_0:!
18850    RPLOT -5,16
18855    RPLOT 5,16
18860    RPLOT 5,14
18865    RPLOT 3,14
18870    RPLOT 3,12
18875    RPLOT 5,11
18880    RPLOT 5,0
18885    RPLOT 3,0
```

```
18890     RPLOT -3,0
18895     RPLOT -5,0
18900     RPLOT -5,11
18905     RPLOT -3,12
18910     RPLOT -3,14
18915     RPLOT -5,14
18920     RPLOT -5,16
18925     RETURN
18930 Bottle_1:!
18935     RPLOT -5,16
18940     RPLOT 5,16
18945     RPLOT 5,14
18950     RPLOT 3,14
18955     RPLOT 3,12
18960     RPLOT 5,11
18965     RPLOT 5,0
18970     RPLOT 3,0
18975     RPLOT -3,0
18980     RPLOT -5,0
18985     RPLOT -5,11
18990     RPLOT -3,12
18995     RPLOT -3,14
19000     RPLOT -5,14
19005     RPLOT -5,16,2
19010     FOR I=0 TO 7
19015        RPLOT -5,I
19020        RPLOT 5,I,2
19025     NEXT I
19030     RETURN
19035 Cup_0:!
19040     RPLOT -3,24
19045     RPLOT 3,24
19050     RPLOT 3,18
19055     RPLOT 8,0
19060     RPLOT -8,0
19065     RPLOT -3,18
19070     RPLOT -3,24
19075     RETURN
19080 Cup_1:!
19085     RPLOT -3,24
19090     RPLOT 3,24
19095     RPLOT 3,18
19100     RPLOT 8,0
19105     RPLOT -8,0
19110     RPLOT -3,18
19115     RPLOT -3,24,2
19120     FOR I=0 TO 12
19125        RPLOT -8+(I DIV 3),I
19130        RPLOT 8-(I DIV 3),I,2
19135     NEXT I
19140     RETURN
19145 Cup_2:!
19150     RPLOT -3,24
19155     RPLOT 3,24
19160     RPLOT 3,18
19165     RPLOT 8,0
19170     RPLOT -8,0
19175     RPLOT -3,18
19180     RPLOT -3,24,2
19185     FOR I=0 TO 12
19190        SELECT I
19195        CASE 0,1,3
19200           RPLOT -8+(I DIV 3),I
19205           RPLOT 8-(I DIV 3),I,2
19210        END SELECT
19215     NEXT I
19220     RETURN
19225 Titrater:!
19230     RPLOT -20,0
19235     RPLOT -20,15
19240     RPLOT 0,15
19245     RPLOT 10,5
```

```
19250     RPLOT 20,5
19255     RPLOT 20,0
19260     RPLOT -20,0,2
19265     RPLOT -14,15
19270     RPLOT -14,35
19275     RPLOT -11,35
19280     RPLOT -11,38
19285     RPLOT -9,38
19290     RPLOT -9,35
19295     RPLOT -6,35
19300     RPLOT -6,15
19305     RETURN
19310 Tenbin:!
19315     RPLOT -18,0
19320     RPLOT -18,23
19325     RPLOT 18,23
19330     RPLOT 18,0
19335     RPLOT -18,0,2
19340     RPLOT -18,4
19345     RPLOT 18,4
19350     RPLOT 18,7
19355     RPLOT -18,7,2
19360     RPLOT -3,23
19365     RPLOT -3,30
19370     RPLOT -18,30
19375     RPLOT -18,33
19380     RPLOT 18,33
19385     RPLOT 18,30
19390     RPLOT -3,30
19395     RPLOT 3,30
19400     RPLOT 3,23
19405     RETURN
19410 Stirer:!
19415     RPLOT 0,33
19420     RPLOT 0,5
19425     RPLOT -4,5
19430     RPLOT 4,5
19435     RETURN
19440 B_table:!
19445     RPLOT -19,16
19450     RPLOT -19,24
19455     RPLOT -7,24
19460     RPLOT -7,28
19465     RPLOT 7,28
19470     RPLOT 7,24
19475     RPLOT 19,24
19480     RPLOT 19,16,2
19485     RPLOT -31,16
19490     RPLOT 31,16
19495     RPLOT 31,12
19500     RPLOT -31,12
19505     RPLOT -31,16,2
19510     RPLOT -31,12
19515     RPLOT -31,4,2
19520     RPLOT 31,12
19525     RPLOT 31,4,2
19530     RPLOT -31,4
19535     RPLOT 31,4
19540     RPLOT 31,0
19545     RPLOT -31,0
19550     RPLOT -31,4
19555     RETURN
19560 C_table:!
19565     RPLOT -22,0
19570     RPLOT -22,20
19575     RPLOT -7,20
19580     RPLOT -7,24
19585     RPLOT 7,24
19590     RPLOT 7,20
19595     RPLOT 22,20
19600     RPLOT 22,0
19605     RPLOT -22,0,2
```

```
19610    RPLOT -22,4
19615    RPLOT 22,4,2
19620    RPLOT -22,16
19625    RPLOT 22,16,2
19630 SUBEND
19635 !
19640 SUB Check_data(INTEGER Ret_code)
19645 !*****************************************************
19650 !*
19655 !*      Check  the data and Disk
19660 !*
19665 !*****************************************************
19670    OPTION BASE 1
19675    COM /Chit_data/ Unit$(*),Chit$(*),Name$(*),Date$(*)
19680    COM /Sample_1/ REAL S1_sample(*),S1_dilute(*)
19685    COM /Sample_2/ INTEGER S2_cup(*),S2_numbers(*),REAL S2_sample(*)
19690    COM /Index_data/ INTEGER File_num,Idx_chit$(*)
19695    COM /Index_data/ Idx_unit$(*),Idx_name$(*),Idx_date$(*)
19700    COM /Measure/ INTEGER Solution,Sampling,Max_cup,Max_bottle
19705    COM /Measure/ INTEGER Now_cup,Now_bottle,S2_count
19710    INTEGER I,J,K,Cup
19715    REAL Max,Temp
19720    !
19725 Check_data:!
19730    Ret_code=0
19735    FOR I=1 TO 2
19740       FOR J=1 TO 8
19745          SELECT NUM(Chit$(I)[J;1])
19750          CASE 0 TO 31,33 TO 47
19755             Ret_code=5
19760          CASE 58 TO 64
19765             Ret_code=5
19770          CASE 91 TO 94,96
19775             Ret_code=5
19780          CASE 123 TO 127
19785             Ret_code=5
19790          CASE 128 TO 160,255
19795             Ret_code=5
19800          END SELECT
19805       NEXT J
19810    NEXT I
19815    IF Chit$(1)=RPT$(" ",8) AND Chit$(2)=RPT$(" ",8) THEN
19820       Ret_code=4
19825    END IF
19830    IF Ret_code<>0 THEN SUBEXIT
19835    !
19840 Chk_disk:!
19845    CALL Get_date(Date$(1))
19850    CALL Get_date(Date$(2))
19855    Max_file=140
19860    CALL Get_index(Ret_code)
19865    IF Ret_code=0 THEN
19870       IF File_num>Max_file THEN
19875          Ret_code=11
19880       ELSE
19885          FOR I=1 TO 2
19890             FOR J=1 TO File_num
19895                IF Chit$(I)=Idx_chit$(J) THEN
19900                   Ret_code=5
19905                END IF
19910             NEXT J
19915          NEXT I
19920       END IF
19925    END IF
19930 SUBEND
19935 !
19940 SUB Disp_index(INTEGER Page)
19945 !*********************************************
19950 !*
19955 !*     Display the Index of File
19960 !*
19965 !*********************************************
```

```
19970     OPTION BASE 1
19975     COM /Index_data/ INTEGER File_num
19980     COM /Index_data/ Idx_chit$(*),Idx_unit$(*)
19985     COM /Index_data/ Idx_name$(*),Idx_date$(*)
19990     DIM Guide$[40],Buf$[20]
19995     INTEGER I,X,Y,No,Max_page
20000     !
20005 Disp_index:!
20010     Max_page=((File_num-1) DIV 24)+1
20015     Page=((Page-1) MOD Max_page)+1
20020     OUTPUT 2;"K";
20025     GCLEAR
20030    !Guide$="3JG<%U%!%$%k!=Mw"
20035    !CALL K_move(40,92)
20040    !CALL K_label(Guide$)
20045     PRINT TABXY(36,1);"File  List"
20050     PRINT TABXY(67,2);"page "&VAL$(Page)&"/"&VAL$(Max_page)
20055     FOR I=1 TO 2
20060        MOVE 200*(I-1)+7,258
20065        GOSUB Frame_1
20070        PRINT TABXY(40*(I-1)+4,5);"No.       Date          Report No."
20075     NEXT I
20080     FOR I=1 TO 24
20085        X=40*((I-1) DIV 12)
20090        Y=((I-1) MOD 12)+7
20095        No=I+24*(Page-1)
20100        IF No<=File_num THEN
20105           PRINT TABXY(X+4,Y);RPT$(" ",3-LEN(VAL$(No)))&VAL$(No)
20110           PRINT TABXY(X+11,Y);"19"&Idx_date$(No)
20115           PRINT TABXY(X+28,Y);Idx_chit$(No)
20120        END IF
20125     NEXT I
20130     SUBEXIT
20135 !
20140 !----- SUBROUTINES
20145 !
20150 Frame_1:!
20155     RPLOT 0,0
20160     RPLOT 185,0
20165     RPLOT 185,-178
20170     RPLOT 0,-178
20175     RPLOT 0,0
20180     RPLOT 0,-24
20185     RPLOT 185,-24,2
20190     RPLOT 110,0
20195     RPLOT 110,-178
20200     RPLOT 30,-178
20205     RPLOT 30,0
20210     RETURN
20215 SUBEND
20220 !
20225 SUB Disp_result(INTEGER Page)
20230 !*********************************************
20235 !*
20240 !*     Display the Result of Measurement
20245 !*
20250 !*********************************************
20255     OPTION BASE 1
20260     COM /File_data/ Chit_x$,Unit_x$,Name_x$,Date_x$
20265     COM /File_data/ REAL Sample_1(*),Dilute(*)
20270     COM /File_data/ REAL Sample_2(*),INTEGER Cup_num(*),Bottles$(*)
20275     COM /File_data/ REAL Inp_vials(*),Mea_vials(*)
20280     COM /Get_data/ INTEGER Max_page,Max_data
20285     COM /Get_data/ REAL Sample_1x,Dilute_x,Dilute_rate
20290     COM /Get_data/ REAL Sample_2x(*),Bottle_x$(*)
20295     DIM Guide$[40],Inp$[20]
20300     INTEGER I,X,Y,No
20305     INTEGER Temp1,Temp2
20310     REAL Temp3
20315     !
20320 Disp_result:!
20325     Max_page=0
```

```
20330    CALL Get_page(Max_page)
20335    Page=((Page-1) MOD Max_page)+1
20340    CALL Get_page(Page)
20345    OUTPUT 2;"K";
20350    GCLEAR
20355   !Guide$="(+FO=ENL%5%s%W%j%s%OAuCVGiNL7k2L"
20360   !CALL K_move(17,92)
20365   !CALL K_label(Guide$)
20370    PRINT TABXY(5,1);"Automated Gravimetric Sampling System      Measuremen
t Result"
20375    PRINT TABXY(70,3);"page "&VAL$(Page)&"/"&VAL$(Max_page)
20380    MOVE 7,258
20385    GOSUB Frame_1
20390    PRINT TABXY(3,5);" Unit No.        ";Unit_x$
20395    Temp3=INT(Sample_1x*10000+.5)/10000
20400    PRINT TABXY(3,6);" 1st Sampling    ";Temp3
20405    Temp3=INT(Dilute_rate*10000+.5)/10000
20410    PRINT TABXY(3,7);" Dilute-Factor   ";Temp3
20415    MOVE 152,258
20420    GOSUB Frame_1
20425    PRINT TABXY(32,5);" Report No.       ";Chit_x$
20430    Temp3=INT(Dilute_x*10000+.5)/10000
20435    PRINT TABXY(32,6);" Dilution (g)    ";Temp3
20440    PRINT TABXY(32,7);" Operator        ";Name_x$
20445    PRINT TABXY(61,6);"Date : ";"19"&Date_x$
20450    !
20455    FOR I=1 TO 3
20460       MOVE 130*(I-1)+7,198
20465       GOSUB Frame_2
20470       PRINT TABXY(26*(I-1)+4,10);"No.   Vial   sample (g)"
20475    NEXT I
20480    FOR No=1 TO 20
20485       X=26*((No-1) DIV 7)
20490       Y=((No-1) MOD 7)+12
20495       PRINT TABXY(X+4,Y);RPT$(" ",2-LEN(VAL$(No)))&VAL$(No)
20500       IF Max_data<No THEN
20505          PRINT TABXY(X+9,Y);RPT$(" ",8)
20510          PRINT TABXY(X+17,Y);RPT$(" ",10)
20515       ELSE
20520          PRINT TABXY(X+8,Y);" "&Bottle_x$(No)&" "
20525          Temp3=INT(Sample_2x(No)*10000+.5)/10000
20530          PRINT TABXY(X+17,Y);RPT$(" ",9-LEN(VAL$(Temp3)))&VAL$(Temp3)
20535       END IF
20540    NEXT No
20545    SUBEXIT
20550 !
20555 !----- SUBROUTINES
20560 !
20565 Frame_1:!
20570    RPLOT 0,0
20575    RPLOT 140,0
20580    RPLOT 140,-48
20585    RPLOT 0,-48
20590    RPLOT 0,0
20595    RPLOT 75,0
20600    RPLOT 75,-48
20605    RETURN
20610 Frame_2:!
20615    RPLOT 0,0
20620    RPLOT 125,0
20625    RPLOT 125,-118
20630    RPLOT 0,-118
20635    RPLOT 0,0
20640    RPLOT 0,-24
20645    RPLOT 125,-24,2
20650    RPLOT 70,0
20655    RPLOT 70,-118
20660    RPLOT 25,-118
20665    RPLOT 25,0
20670    RETURN
20675 SUBEND
20680 !
```

```
20685 SUB Print_result(INTEGER Ret_code)
20690 !************************************************
20695 !*
20700 !*      Print  Result of Measurement
20705 !*
20710 !************************************************
20715    OPTION BASE 1
20720    COM /File_data/ Chit_x$,Unit_x$,Name_x$,Date_x$
20725    COM /File_data/ REAL Sample_1(*),Dilute(*)
20730    COM /File_data/ REAL Sample_2(*),INTEGER Cup_num(*),Bottles$(*)
20735    COM /File_data/ REAL Inp_vials(*),Mea_vials(*)
20740    COM /Get_data/ INTEGER Max_page,Max_data
20745    COM /Get_data/ REAL Sample_1x,Dilute_x,Dilute_rate
20750    COM /Get_data/ REAL Sample_2x(*),Bottle_x$(*)
20755    DIM Buf$[32],Buf1$[32],Buf2$[32],Buf3$[32]
20760    DIM Buf4$[32],Buf5$[32],Buf6$[32]
20765    INTEGER I,J,K,Page
20770    INTEGER Temp1,Temp2
20775    REAL Temp3,Bias(20),Samp_1(20),Dilut_1(20),Dilut_f(20)
20780    !
20785 Print_result:!
20790    PRINTER IS 30;WIDTH 300
20795    PRINT CHR$(27)&"C"&CHR$(6);
20800    PRINT CHR$(27)&"I"&CHR$(136);
20805    PRINT CHR$(27)&"2"&CHR$(0)&CHR$(10);
20810    PRINT CHR$(27)&"8"&CHR$(0)&CHR$(0);
20815    PRINT
20820    PRINT
20825    PRINT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!";
20830    PRINT "#M#e#a#s#u#r#e#m#e#n#t!!#R#e#s#u#l#t"
20835    PRINT
20840    PRINT
20845    PRINT CHR$(27)&"I"&CHR$(136);
20850    PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4);
20855    GOSUB L_margin
20860    PRINT RPT$("/'",28);
20865    PRINT CHR$(27)&"8"&CHR$(0)&CHR$(0)&"/'/'.S.i.g.n.a.t.u.r.e/'";
20870    PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4);
20875    PRINT RPT$("/'",1)&".N.S.B/'/'/'";
20880    PRINT CHR$(27)&"I"&CHR$(137);RPT$("/'",20)
20885    PRINT CHR$(27)&"I"&CHR$(136)
20890    GOSUB L_margin
20895    PRINT RPT$("/'",38)&".I.A.E.A/'/'";
20900    PRINT CHR$(27)&"I"&CHR$(137);RPT$("/'",20)
20905    PRINT CHR$(27)&"I"&CHR$(136)
20910    PRINT
20915    GOSUB L_margin
20920    PRINT RPT$("/'",38);".D.a.t.e/'/'";
20925    Buf$="  19"&Date_x$[1;2]&" "
20930    CALL Kanji_conv(Buf$,Buf1$)
20935    Buf$=" "&Date_x$[4;2]&" "
20940    CALL Kanji_conv(Buf$,Buf2$)
20945    Buf$=" "&Date_x$[7;2]&" "
20950    CALL Kanji_conv(Buf$,Buf3$)
20955    PRINT Buf1$;".-";Buf2$;".-";Buf3$
20960    PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4)
20965    PRINT CHR$(27)&"C"&CHR$(12);
20970    GOSUB L_margin
20975    GOSUB Line_11
20980    GOSUB Line_11
20985    PRINT
20990    FOR I=1 TO 2
20995       FOR J=1 TO 4
21000          GOSUB L_margin
21005          FOR K=1 TO 2
21010             SELECT J
21015             CASE 1,3
21020                GOSUB Line_12
21025             CASE 2
21030                SELECT 2*(K-1)+I
21035                CASE 1
21040                   Buf$="  "&Unit_x$&"    "
```

```
21045                   CALL Kanji_conv(Buf$,Buf1$)
21050                   GOSUB Line_data11
21055                CASE 2
21060                   Buf$=RPT$(" ",15)
21065                   CALL Kanji_conv(Buf$,Buf1$)
21070                   GOSUB Line_data12
21075                CASE 3
21080                   Buf$="   "&Chit_x$&"    "
21085                   CALL Kanji_conv(Buf$,Buf1$)
21090                   GOSUB Line_data14
21095                CASE 4
21100                   Buf$="  "&Name_x$&"   "
21105                   CALL Kanji_conv(Buf$,Buf1$)
21110                   GOSUB Line_data16
21115                END SELECT
21120             CASE 4
21125                IF I=2 THEN
21130                   GOSUB Line_14
21135                ELSE
21140                   GOSUB Line_13
21145                END IF
21150             END SELECT
21155          NEXT K
21160          PRINT
21165       NEXT J
21170    NEXT I
21175    PRINT
21180    FOR I=1 TO 20
21185       FOR J=1 TO 4
21190          IF Cup_num(I)=J THEN
21195             Samp_1(I)=Sample_1(J)
21200             Dilut_1(I)=Dilute(J)
21205          END IF
21210       NEXT J
21215    NEXT I
21220    FOR J=1 TO 5
21225       GOSUB L_margin
21230       SELECT J
21235       CASE 1
21240          GOSUB Line_31
21245       CASE 2,4
21250          GOSUB Line_32
21255       CASE 3
21260          GOSUB Line_menu31
21265       CASE 5
21270          GOSUB Line_33
21275       END SELECT
21280       PRINT
21285    NEXT J
21290    FOR I=1 TO 20
21295       FOR J=1 TO 4
21300          GOSUB L_margin
21305          SELECT J
21310             CASE 1,3
21315                GOSUB Line_32
21320             CASE 2
21325                Buf$=RPT$(" ",3-LEN(VAL$(I)))&VAL$(I)&" "
21330                CALL Kanji_conv(Buf$,Buf1$)
21335                IF Sample_2(I)>0 THEN
21340                   Buf$=RPT$(" ",3)&Bottles$(I)&RPT$(" ",3)
21345                   CALL Kanji_conv(Buf$,Buf2$)
21350                ELSE
21355                   Buf2$=RPT$("/'",12)
21360                END IF
21365                IF Samp_1(I)>0 THEN
21370                   Temp3=INT(Samp_1(I)*10000+.5)/10000
21375                   Buf$=RPT$(" ",10-LEN(VAL$(Temp3)))&VAL$(Temp3)&"  "
21380                   CALL Kanji_conv(Buf$,Buf3$)
21385                ELSE
21390                   Buf3$=RPT$("/'",12)
21395                END IF
21400                IF Dilut_1(I)>0 THEN
```

```
21405                   Temp3=INT(Dilut_1(I)*10000+.5)/10000
21410                   Buf$=RPT$(" ",10-LEN(VAL$(Temp3)))&VAL$(Temp3)&"  "
21415                   CALL Kanji_conv(Buf$,Buf4$)
21420                ELSE
21425                   Buf4$=RPT$("/",12)
21430                END IF
21435                IF Samp_1(I)>0 THEN
21440                   Dilut_f(I)=Dilut_1(I)/Samp_1(I)
21445                   Temp3=INT(Dilut_f(I)*10000+.5)/10000
21450                   Buf$=RPT$(" ",10-LEN(VAL$(Temp3)))&VAL$(Temp3)&"  "
21455                   CALL Kanji_conv(Buf$,Buf5$)
21460                ELSE
21465                   Buf5$=RPT$("/",12)
21470                END IF
21475                IF Sample_2(I)>0 THEN
21480                   Temp3=INT(Sample_2(I)*10000+.5)/10000
21485                   Buf$=RPT$(" ",10-LEN(VAL$(Temp3)))&VAL$(Temp3)&"  "
21490                   CALL Kanji_conv(Buf$,Buf6$)
21495                ELSE
21500                   Buf6$=RPT$("/",12)
21505                END IF
21510                GOSUB Line_data3
21515             CASE 3
21520                GOSUB Line_32
21525             CASE 4
21530                IF I=20 THEN
21535                   GOSUB Line_34
21540                ELSE
21545                   GOSUB Line_33
21550                END IF
21555             END SELECT
21560          PRINT
21565       NEXT J
21570    NEXT I
21575    PRINT CHR$(12);
21580    PRINT
21585    PRINT
21590    PRINT
21595    PRINT
21600    PRINT
21605    PRINT
21610    PRINT CHR$(27)&"C"&CHR$(6);
21615    PRINT CHR$(27)&"I"&CHR$(136);
21620    PRINT CHR$(27)&"2"&CHR$(0)&CHR$(10);
21625    PRINT CHR$(27)&"8"&CHR$(0)&CHR$(0);
21630    PRINT
21635    PRINT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!";
21640    PRINT "#A#u#t#h#e#n#t#i#c#a#t#i#o#n!!#R#e#s#u#l#t"
21645    PRINT
21650    PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4)
21655    PRINT CHR$(27)&"C"&CHR$(12);
21660    GOSUB L_margin
21665    PRINT
21670    PRINT
21675    FOR J=1 TO 5
21680       GOSUB L_margin
21685       SELECT J
21690       CASE 1
21695          GOSUB Line_41
21700       CASE 2,4
21705          GOSUB Line_42
21710       CASE 3
21715          GOSUB Line_menu41
21720       CASE 5
21725          GOSUB Line_43
21730       END SELECT
21735       PRINT
21740    NEXT J
21745    FOR I=1 TO 20
21750       FOR J=1 TO 4
21755          GOSUB L_margin
21760          SELECT J
```

```
21765          CASE 1,3
21770             GOSUB Line_42
21775          CASE 2
21780             Buf$=RPT$(" ",3-LEN(VAL$(I)))&VAL$(I)&" "
21785             CALL Kanji_conv(Buf$,Buf1$)
21790             IF (Mea_vials(I)>0) OR (Inp_vials(I)>0) THEN
21795                Buf$=RPT$(" ",4)&Bottles$(I)&RPT$(" ",4)
21800                CALL Kanji_conv(Buf$,Buf2$)
21805             ELSE
21810                Buf2$=RPT$("/'",14)
21815             END IF
21820             IF Inp_vials(I)>0 THEN
21825                Temp3=INT(Inp_vials(I)*10000+.5)/10000
21830                Buf$=RPT$(" ",11-LEN(VAL$(Temp3)))&VAL$(Temp3)&"    "
21835                CALL Kanji_conv(Buf$,Buf3$)
21840             ELSE
21845                Buf3$=RPT$("/'",15)
21850             END IF
21855             IF Mea_vials(I)>0 THEN
21860                Temp3=INT(Mea_vials(I)*10000+.5)/10000
21865                Buf$=RPT$(" ",11-LEN(VAL$(Temp3)))&VAL$(Temp3)&"    "
21870                CALL Kanji_conv(Buf$,Buf4$)
21875             ELSE
21880                Buf4$=RPT$("/'",15)
21885             END IF
21890             IF (Inp_vials(I)>0) AND (Mea_vials(I)>0) THEN
21895                Bias(I)=Inp_vials(I)-Mea_vials(I)
21900                Temp3=INT(Bias(I)*10000+.5)/10000
21905                Buf$=RPT$(" ",11-LEN(VAL$(Temp3)))&VAL$(Temp3)&"    "
21910                CALL Kanji_conv(Buf$,Buf5$)
21915             ELSE
21920                Buf5$=RPT$("/'",15)
21925             END IF
21930             GOSUB Line_data4
21935          CASE 3
21940             GOSUB Line_42
21945          CASE 4
21950             IF I=20 THEN
21955                GOSUB Line_44
21960             ELSE
21965                GOSUB Line_43
21970             END IF
21975          END SELECT
21980          PRINT
21985       NEXT J
21990    NEXT I
21995    PRINT CHR$(12);
22000    PRINTER IS 1
22005    SUBEXIT
22010  !
22015  !---------- SUBROUTINES
22020  !
22025 L_margin:!
22030    PRINT "!!";
22035    RETURN
22040 Line_data11:!
22045    PRINT CHR$(27)&"H"&CHR$(5);
22050    PRINT CHR$(47)&CHR$(122);
22055    PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4);
22060    PRINT "/'/'/'.U.n.i.t/'.N.o../'/'/'/'/'/'";
22065    PRINT CHR$(47)&CHR$(122)&Buf1$;
22070    PRINT CHR$(47)&CHR$(122);
22075    RETURN
22080 Line_data12:!
22085    PRINT CHR$(27)&"H"&CHR$(5);
22090    PRINT CHR$(47)&CHR$(122);
22095    PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4);
22100    PRINT "/'. . . /'. . . . . . . /'. . . ";
22105    PRINT CHR$(47)&CHR$(122)&Buf1$;
22110    PRINT CHR$(47)&CHR$(122);
22115    RETURN
22120 Line_data14:!
```

```
22125     PRINT CHR$(27)&"H"&CHR$(5);
22130     PRINT CHR$(47)&CHR$(122);
22135     PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4);
22140     PRINT "/'/'.R.e.p.o.r.t/'.N.o../'/'/'/'/'";
22145     PRINT CHR$(47)&CHR$(122)&Buf1$;
22150     PRINT CHR$(47)&CHR$(122);
22155     RETURN
22160 Line_data16:!
22165     PRINT CHR$(27)&"H"&CHR$(5);
22170     PRINT CHR$(47)&CHR$(122);
22175     PRINT CHR$(27)&"8"&CHR$(4)&CHR$(4);
22180     PRINT "/'/'.O.p.e.r.a.t.o.r/'/'/'/'/'/'";
22185     PRINT CHR$(47)&CHR$(122)&Buf1$;
22190     PRINT CHR$(47)&CHR$(122);
22195     RETURN
22200 Line_11:!
22205     PRINT CHR$(27)&"H"&CHR$(5);
22210     PRINT CHR$(47)&CHR$(100)&RPT$(CHR$(47)&CHR$(99),17);
22215     PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),15);
22220     PRINT CHR$(47)&CHR$(101);
22225     RETURN
22230 Line_12:!
22235     PRINT CHR$(27)&"H"&CHR$(5);
22240     PRINT CHR$(47)&CHR$(122)&RPT$("/'",17);
22245     PRINT CHR$(47)&CHR$(122)&RPT$("/'",15);
22250     PRINT CHR$(47)&CHR$(122);
22255     RETURN
22260 Line_13:!
22265     PRINT CHR$(27)&"H"&CHR$(5);
22270     PRINT CHR$(47)&CHR$(107)&RPT$(CHR$(47)&CHR$(99),17);
22275     PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),15);
22280     PRINT CHR$(47)&CHR$(105);
22285     RETURN
22290 Line_14:!
22295     PRINT CHR$(27)&"H"&CHR$(5);
22300     PRINT CHR$(47)&CHR$(103)&RPT$(CHR$(47)&CHR$(99),17);
22305     PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),15);
22310     PRINT CHR$(47)&CHR$(102);
22315     RETURN
22320     !
22325 Line_menu31:!
22330     PRINT CHR$(27)&"H"&CHR$(5);
22335     PRINT CHR$(47)&CHR$(122)&"/'.N.o..";
22340     PRINT CHR$(47)&CHR$(122);CHR$(27)&"8"&CHR$(4)&CHR$(4);
22345     PRINT "/'/'.V.i.a.l/'.N.o../'/'"&CHR$(47)&CHR$(122);
22350     PRINT "/'.S.a.m.p.-.1/'.(.g.)/'"&CHR$(47)&CHR$(122);
22355     PRINT "/'.D.i.l.u.t.e/'.(.g.)/'"&CHR$(47)&CHR$(122);
22360     PRINT "/'.D.i.l.u.t.e/'. .F. /'"&CHR$(47)&CHR$(122);
22365     PRINT "/'.S.a.m.p.-.2/'.(.g.)/'"&CHR$(47)&CHR$(122);
22370     RETURN
22375 Line_data3:!
22380     PRINT CHR$(27)&"H"&CHR$(5);
22385     PRINT CHR$(47)&CHR$(122)&Buf1$;
22390     PRINT CHR$(47)&CHR$(122)&Buf2$;
22395     PRINT CHR$(47)&CHR$(122)&Buf3$;
22400     PRINT CHR$(47)&CHR$(122)&Buf4$;
22405     PRINT CHR$(47)&CHR$(122)&Buf5$;
22410     PRINT CHR$(47)&CHR$(122)&Buf6$;
22415     PRINT CHR$(47)&CHR$(122);
22420     RETURN
22425 Line_31:!
22430     PRINT CHR$(27)&"H"&CHR$(5);
22435     PRINT CHR$(47)&CHR$(100)&RPT$(CHR$(47)&CHR$(99),4);
22440     PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),12);
22445     PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),12);
22450     PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),12);
22455     PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),12);
22460     PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),12);
22465     PRINT CHR$(47)&CHR$(101);
22470     RETURN
22475 Line_32:!
22480     PRINT CHR$(27)&"H"&CHR$(5);
```

```
22485     PRINT CHR$(47)&CHR$(122)&RPT$("/'",4);
22490     PRINT CHR$(47)&CHR$(122)&RPT$("/'",12);
22495     PRINT CHR$(47)&CHR$(122)&RPT$("/'",12);
22500     PRINT CHR$(47)&CHR$(122)&RPT$("/'",12);
22505     PRINT CHR$(47)&CHR$(122)&RPT$("/'",12);
22510     PRINT CHR$(47)&CHR$(122)&RPT$("/'",12);
22515     PRINT CHR$(47)&CHR$(122);
22520     RETURN
22525 Line_33:!
22530     PRINT CHR$(27)&"H"&CHR$(5);
22535     PRINT CHR$(47)&CHR$(107)&RPT$(CHR$(47)&CHR$(99),4);
22540     PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),12);
22545     PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),12);
22550     PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),12);
22555     PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),12);
22560     PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),12);
22565     PRINT CHR$(47)&CHR$(105);
22570     RETURN
22575 Line_34:!
22580     PRINT CHR$(27)&"H"&CHR$(5);
22585     PRINT CHR$(47)&CHR$(103)&RPT$(CHR$(47)&CHR$(99),4);
22590     PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),12);
22595     PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),12);
22600     PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),12);
22605     PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),12);
22610     PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),12);
22615     PRINT CHR$(47)&CHR$(102);
22620     RETURN
22625 Line_menu41:!
22630     PRINT CHR$(27)&"H"&CHR$(5);
22635     PRINT CHR$(47)&CHR$(122)&"/',N.o..";
22640     PRINT CHR$(47)&CHR$(122);CHR$(27)&"8"&CHR$(4)&CHR$(4);
22645     PRINT "/'/' .V.i.a.l/'.N.o.. /'/'"&CHR$(47)&CHR$(122);
22650     PRINT "/'.I.n.p.u.t.-.D.a.t.a.(.g.)/'"&CHR$(47)&CHR$(122);
22655     PRINT ".M.e.a.s...(.T.a.r.e.).(.g.)/'"&CHR$(47)&CHR$(122);
22660     PRINT "/'. . .B.i.a.s. . . /'.(.g.)/'"&CHR$(47)&CHR$(122);
22665     RETURN
22670 Line_data4:!
22675     PRINT CHR$(27)&"H"&CHR$(5);
22680     PRINT CHR$(47)&CHR$(122)&Buf1$;
22685     PRINT CHR$(47)&CHR$(122)&Buf2$;
22690     PRINT CHR$(47)&CHR$(122)&Buf3$;
22695     PRINT CHR$(47)&CHR$(122)&Buf4$;
22700     PRINT CHR$(47)&CHR$(122)&Buf5$;
22705     PRINT CHR$(47)&CHR$(122);
22710     RETURN
22715 Line_41:!
22720     PRINT CHR$(27)&"H"&CHR$(5);
22725     PRINT CHR$(47)&CHR$(100)&RPT$(CHR$(47)&CHR$(99),4);
22730     PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),14);
22735     PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),15);
22740     PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),15);
22745     PRINT CHR$(47)&CHR$(104)&RPT$(CHR$(47)&CHR$(99),15);
22750     PRINT CHR$(47)&CHR$(101);
22755     RETURN
22760 Line_42:!
22765     PRINT CHR$(27)&"H"&CHR$(5);
22770     PRINT CHR$(47)&CHR$(122)&RPT$("/'",4);
22775     PRINT CHR$(47)&CHR$(122)&RPT$("/'",14);
22780     PRINT CHR$(47)&CHR$(122)&RPT$("/'",15);
22785     PRINT CHR$(47)&CHR$(122)&RPT$("/'",15);
22790     PRINT CHR$(47)&CHR$(122)&RPT$("/'",15);
22795     PRINT CHR$(47)&CHR$(122);
22800     RETURN
22805 Line_43:!
22810     PRINT CHR$(27)&"H"&CHR$(5);
22815     PRINT CHR$(47)&CHR$(107)&RPT$(CHR$(47)&CHR$(99),4);
22820     PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),14);
22825     PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),15);
22830     PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),15);
22835     PRINT CHR$(47)&CHR$(108)&RPT$(CHR$(47)&CHR$(99),15);
22840     PRINT CHR$(47)&CHR$(105);
```

```
22845     RETURN
22850 Line_44:!
22855     PRINT CHR$(27)&"H"&CHR$(5);
22860     PRINT CHR$(47)&CHR$(103)&RPT$(CHR$(47)&CHR$(99),4);
22865     PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),14);
22870     PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),15);
22875     PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),15);
22880     PRINT CHR$(47)&CHR$(106)&RPT$(CHR$(47)&CHR$(99),15);
22885     PRINT CHR$(47)&CHR$(102);
22890     RETURN
22895 SUBEND
22900 !
22905 SUB Get_page(INTEGER Page)
22910 !*****************************************
22915 !*
22920 !*     Get result-data ( 1 page )
22925 !*
22930 !*****************************************
22935     OPTION BASE 1
22940     COM /File_data/ Chit_x$(*),Unit_x$,Name_x$,Date_x$
22945     COM /File_data/ REAL Sample_1(*),Dilute(*)
22950     COM /File_data/ REAL Sample_2(*),INTEGER Cup_num(*),Bottles$(*)
22955     COM /File_data/ REAL Inp_vials(*),Mea_vials(*)
22960     COM /Get_data/ INTEGER Max_page,Max_data
22965     COM /Get_data/ REAL Sample_1x,Dilute_x,Dilute_rate
22970     COM /Get_data/ REAL Sample_2x(*),Bottle_x$(*)
22975     INTEGER I,N,Cup
22980     !.
22985 Get_page:!
22990     IF Page=0 THEN
22995        Max_page=0
23000        FOR I=1 TO 4
23005           IF Sample_1(I)>0 THEN
23010              Max_page=Max_page+1
23015           END IF
23020        NEXT I
23025        Page=Max_page
23030     ELSE
23035        Sample_1x=0
23040        Dilute_x=0
23045        Dilute_rate=0
23050        FOR I=1 TO 20
23055           Sample_2x(I)=0
23060           Bottle_x$(I)=RPT$(" ",6)
23065        NEXT I
23070        N=0
23075        Cup=0
23080        WHILE Page<>N AND Cup<4
23085           Cup=Cup+1
23090           IF Sample_1(Cup)<>0 THEN
23095              N=N+1
23100           END IF
23105        END WHILE
23110        IF Page=N THEN
23115           Sample_1x=Sample_1(Cup)
23120           Dilute_x=Dilute(Cup)
23125           Dilute_rate=INT(10000*(Dilute_x/Sample_1x))/10000
23130           Max_data=0
23135           FOR I=1 TO 20
23140              IF Cup_num(I)=Cup THEN
23145                 Max_data=Max_data+1
23150                 Sample_2x(Max_data)=Sample_2(I)
23155                 Bottle_x$(Max_data)=Bottles$(I)
23160              END IF
23165           NEXT I
23170        END IF
23175     END IF
23180 SUBEND
23185 !
23190 SUB Make_disk(INTEGER Ret_code)
23195 !*************************************************
23200 !*
```

```
23205 !*     Make the User_Disk
23210 !*
23215 !*****************************************************
23220    OPTION BASE 1
23225    DIM Guide$[30],Buf1$[12],Buf2$[46]
23230    INTEGER I,Temp1
23235    REAL R1,R2
23240    !
23245 Make_disk:!
23250    OUTPUT 2;"K";
23255    GCLEAR
23260   !Guide$="%G!<%?MQ%G%#%9%/$N@8@."
23265   !CALL K_move(28,92)
23270   !CALL K_label(Guide$)
23275    PRINT TABXY(27,1);"Format  Disk for Data"
23280    INPUT "Fomatting disk and Create file, OK (Y/N) ? ",Buf$
23285    IF Buf$="Y" THEN
23290       ON ERROR GOTO Error1
23295       DISP "Initializing Disk"
23300       INITIALIZE ":HP9122,700,1"
23305       DISP "Creating File     "
23310       CREATE BDAT "INDEX      :HP9122,700,1",196,52
23315       ASSIGN @File TO "INDEX      :HP9122,700,1"
23320       Temp1=0
23325       Buf2$=RPT$(" ",46)
23330       OUTPUT @File,1;Temp1,Buf$
23335       Buf1$=RPT$(" ",8)
23340       Buf2$=RPT$(" ",10)
23345       FOR I=2 TO 196
23350          OUTPUT @File,I;Buf1$,Buf2$,Buf2$,Buf1$
23355       NEXT I
23360       ASSIGN @File TO *
23365       DISP "                                  "
23370    END IF
23375    OFF ERROR
23380    Ret_code=0
23385    SUBEXIT
23390    !
23395 Error1:!
23400    I=ERRN
23405    CALL Set_errn(I,Ret_code)
23410    OFF ERROR
23415    ASSIGN @File TO *
23420 SUBEND
23425 !
23430 SUB Get_index(INTEGER Ret_code)
23435 !*************************************
23440 !*
23445 !*    Read the Index-file
23450 !*
23455 !*************************************
23460    OPTION BASE 1
23465    COM /Index_data/ INTEGER File_num,Idx_chit$(*)
23470    COM /Index_data/ Idx_unit$(*),Idx_name$(*),Idx_date$(*)
23475    INTEGER I
23480    !
23485 Get_index:!
23490    ON ERROR GOTO Error1
23495    ASSIGN @File TO "INDEX      :HP9122,700,1"
23500    ENTER @File,1;File_num
23505    FOR I=1 TO File_num
23510       ENTER @File,I+1;Idx_chit$(I),Idx_unit$(I),Idx_name$(I),Idx_date$(I)
23515    NEXT I
23520    OFF ERROR
23525    ASSIGN @File TO *
23530    Ret_code=0
23535    SUBEXIT
23540    !
23545 Error1:!
23550    I=ERRN
23555    CALL Set_errn(I,Ret_code)
23560    OFF ERROR
```

```
23565    ASSIGN @File TO *
23570 SUBEND
23575 !
23580 SUB Put_index(INTEGER Ret_code)
23585 !*****************************************
23590 !*
23595 !*    Write the Index-File
23600 !*
23605 !*****************************************
23610    OPTION BASE 1
23615    COM /Index_data/ INTEGER File_num,Idx_chit$(*)
23620    COM /Index_data/ Idx_unit$(*),Idx_name$(*),Idx_date$(*)
23625    INTEGER I
23630    !
23635 Put_index:!
23640    ON ERROR GOTO Error1
23645    ASSIGN @File TO "INDEX      :HP9122,700,1"
23650    OUTPUT @File,1;File_num
23655    FOR I=1 TO File_num
23660       OUTPUT @File,I+1;Idx_chit$(I),Idx_unit$(I),Idx_name$(I),Idx_date$(I)
23665    NEXT I
23670    OFF ERROR
23675    ASSIGN @File TO *
23680    Ret_code=0
23685    SUBEXIT
23690    !
23695 Error1:!
23700    I=ERRN
23705    CALL Set_errn(I,Ret_code)
23710    OFF ERROR
23715    ASSIGN @File TO *
23720 SUBEND
23725 !
23730 SUB Get_file(File$,INTEGER Ret_code)
23735 !*****************************************
23740 !*
23745 !*    Read the Data-file
23750 !*
23755 !*****************************************
23760    OPTION BASE 1
23765    COM /Index_data/ INTEGER File_num,Idx_chit$(*)
23770    COM /Index_data/ Idx_unit$(*),Idx_name$(*),Idx_date$(*)
23775    COM /File_data/ Chit_x$,Unit_x$,Name_x$,Date_x$
23780    COM /File_data/ REAL Sample_1(*),Dilute(*)
23785    COM /File_data/ REAL Sample_2(*),INTEGER Cup_num(*),Bottles$(*)
23790    COM /File_data/ REAL Inp_vials(*),Mea_vials(*)
23795    INTEGER I,Flag
23800    !
23805 Get_file:!
23810    I=1
23815    Flag=0
23820    WHILE Flag=0 AND I<=File_num
23825       IF File$=Idx_chit$(I) THEN
23830          Chit_x$=Idx_chit$(I)
23835          Unit_x$=Idx_unit$(I)
23840          Name_x$=Idx_name$(I)
23845          Date_x$=Idx_date$(I)
23850          Flag=1
23855       END IF
23860       I=I+1
23865    END WHILE
23870    IF Flag=1 THEN
23875       ON ERROR GOTO Error1
23880       ASSIGN @File TO File$&":HP9122,700,1"
23885       FOR I=1 TO 4
23890          ENTER @File,I;Sample_1(I),Dilute(I)
23895       NEXT I
23900       FOR I=1 TO 20
23905          ENTER @File,I+4;Bottles$(I),Cup_num(I),Sample_2(I)
23910       NEXT I
23915       FOR I=1 TO 20
23920          ENTER @File,I+24;Inp_vials(I),Mea_vials(I)
```

```
23925       NEXT I
23930       OFF ERROR
23935       ASSIGN @File TO *
23940       Ret_code=0
23945    ELSE
23950       Ret_code=3
23955    END IF
23960    SUBEXIT
23965    !
23970 Error1:!
23975    I=ERRN
23980    CALL Set_errn(I,Ret_code)
23985    OFF ERROR
23990    ASSIGN @File TO *
23995 SUBEND
24000 !
24005 SUB Put_file(INTEGER Solution,Ret_code)
24010 !*****************************************
24015 !*
24020 !*     Write the Data-File
24025 !*
24030 !*****************************************
24035    OPTION BASE 1
24040    COM /Chit_data/ Unit$(*),Chit$(*),Name$(*),Date$(*)
24045    COM /Bottle_data/ Bottle$(*),Inp_vial(*)
24050    COM /Result/ REAL Sample_1st(*),Dilution(*)
24055    COM /Result/ REAL Sample_2nd(*),INTEGER Cup_no(*),REAL Mea_vial(*)
24060    COM /Index_data/ INTEGER File_num,Idx_chit$(*)
24065    COM /Index_data/ Idx_unit$(*),Idx_name$(*),Idx_date$(*)
24070    INTEGER I
24075    !
24080 Put_file:!
24085    Ret_code=0
24090    IF Chit$(Solution)<>RPT$(" ",8) THEN
24095       CALL Get_index(Ret_code)
24100       IF Ret_code=0 THEN
24105          FOR I=1 TO File_num
24110             IF Chit$(Solution)=Idx_chit$(I) THEN
24115                Ret_code=5
24120             END IF
24125          NEXT I
24130          IF Ret_code=0 THEN
24135             ON ERROR GOTO Error1
24140             CREATE BDAT Chit$(Solution)&":HP9122,700,1",48,20
24145             ASSIGN @File TO Chit$(Solution)&":HP9122,700,1"
24150             FOR I=1 TO 4
24155                OUTPUT @File,I;Sample_1st(Solution,I),Dilution(Solution,I)
24160             NEXT I
24165             FOR I=1 TO 20
24170                OUTPUT @File,I+4;Bottle$(I),Cup_no(Solution,I),Sample_2nd(S
olution,I)
24175             NEXT I
24180             FOR I=1 TO 20
24185                OUTPUT @File,I+24;Inp_vial(I),Mea_vial(I)
24190             NEXT I
24195             OUTPUT @File,45;Chit$(Solution)
24200             OUTPUT @File,46;Unit$(Solution)
24205             OUTPUT @File,47;Name$(Solution)
24210             OUTPUT @File,48;Date$(Solution)
24215             OFF ERROR
24220             ASSIGN @File TO *
24225             !
24230             File_num=File_num+1
24235             Idx_chit$(File_num)=Chit$(Solution)
24240             Idx_unit$(File_num)=Unit$(Solution)
24245             Idx_name$(File_num)=Name$(Solution)
24250             Idx_date$(File_num)=Date$(Solution)
24255             CALL Put_index(Ret_code)
24260          END IF
24265       END IF
24270    END IF
24275    SUBEXIT
```

```
24280     !
24285 Error1:!
24290     I=ERRN
24295     CALL Set_errn(I,Ret_code)
24300     OFF ERROR
24305     ASSIGN @File TO *
24310 SUBEND
24315 !
24320 SUB Get_dtfile(INTEGER Ret_code)
24325 !*****************************************
24330 !*
24335 !*
24340 !*
24345 !*****************************************
24350     OPTION BASE 1
24355     COM /Chit_data/ Unit$(*),Chit$(*),Name$(*),Date$(*)
24360     COM /Sample_1/ REAL S1_sample(*),S1_dilute(*)
24365     COM /Sample_2/ INTEGER S2_cup(*),S2_numbers(*),REAL S2_sample(*)
24370     COM /Bottle_data/ Bottle$(*),Inp_vial(*)
24375     INTEGER I,J
24380    !
24385 Get_dtfile:!
24390     ON ERROR GOTO Error1
24395     ASSIGN @File TO "DTFILE:HP9122,700,0"
24400     FOR I=1 TO 2
24405        ENTER @File;Unit$(I),Chit$(I),Name$(I)
24410        FOR J=1 TO 4
24415           ENTER @File;S1_sample(I,J),S1_dilute(I,J)
24420        NEXT J
24425        FOR J=1 TO 20
24430           ENTER @File;S2_cup(I,J),S2_numbers(I,J),S2_sample(I,J)
24435        NEXT J
24440     NEXT I
24445     FOR I=1 TO 20
24450        ENTER @File;Bottle$(I),Inp_vial(I)
24455     NEXT I
24460     ASSIGN @File TO *
24465     OFF ERROR
24470     FOR I=1 TO 2
24475        Chit$(I)=RPT$(" ",8)
24480        Date$(I)=RPT$(" ",8)
24485     NEXT I
24490     FOR I=1 TO 20
24495        Bottle$(I)=RPT$(" ",6)
24500     NEXT I
24505     SUBEXIT
24510     !
24515 Error1:!
24520     I=ERRN
24525     CALL Set_errn(I,Ret_code)
24530     OFF ERROR
24535     ASSIGN @File TO *
24540 SUBEND
24545 !
24550 SUB Put_dtfile(INTEGER Ret_code)
24555 !*****************************************
24560 !*
24565 !*
24570 !*
24575 !*****************************************
24580     OPTION BASE 1
24585     COM /Chit_data/ Unit$(*),Chit$(*),Name$(*),Date$(*)
24590     COM /Sample_1/ REAL S1_sample(*),S1_dilute(*)
24595     COM /Sample_2/ INTEGER S2_cup(*),S2_numbers(*),REAL S2_sample(*)
24600     COM /Bottle_data/ Bottle$(*),Inp_vial(*)
24605     INTEGER I,J
24610    !
24615 Put_dtfile:!
24620     ON ERROR GOTO Error1
24625     ASSIGN @File TO "DTFILE:HP9122,700,0"
24630     FOR I=1 TO 2
24635        OUTPUT @File;Unit$(I),Chit$(I),Name$(I)
```

```
24640       FOR J=1 TO 4
24645          OUTPUT @File;S1_sample(I,J),S1_dilute(I,J)
24650       NEXT J
24655       FOR J=1 TO 20
24660          OUTPUT @File;S2_cup(I,J),S2_numbers(I,J),S2_sample(I,J)
24665       NEXT J
24670    NEXT I
24675    FOR I=1 TO 20
24680       OUTPUT @File;Bottle$(I),Inp_vial(I)
24685    NEXT I
24690    ASSIGN @File TO *
24695    OFF ERROR
24700    SUBEXIT
24705    !
24710 Error1:!
24715    I=ERRN
24720    CALL Set_errn(I,Ret_code)
24725    OFF ERROR
24730    ASSIGN @File TO *
24735 SUBEND
24740 !
24745 SUB Set_errn(INTEGER Err_code,Ret_code)
24750 !*******************************************
24755 !*
24760 !*     Conversion Error_code
24765 !*
24770 !*******************************************
24775 Set_errn:!
24780    SELECT Err_code
24785    CASE 51,56
24790       Ret_code=10
24795    CASE 54
24800       Ret_code=5
24805    CASE 80
24810       Ret_code=12
24815    CASE ELSE
24820       Ret_code=1
24825    END SELECT
24830 SUBEND
24835 !
24840 SUB Set_date(INTEGER Ret_code)
24845 !*****************************************
24850 !*
24855 !*     Set System-clock
24860 !*
24865 !*****************************************
24870    OPTION BASE 1
24875    COM /Gmax_io/ @Gmax_io
24880    DIM Buf$[17]
24885    DIM Year$[4],Month$[3],Date$[2],Time$[8]
24890    !
24895 Set_date:!
24900    ON TIMEOUT SC(@Gmax_io),5.0 GOTO Timeout
24905    OUTPUT @Gmax_io;"TD"
24910    ENTER @Gmax_io;Buf$
24915    Year$="19"&Buf$[1;2]
24920    Date$=Buf$[7;2]
24925    Time$=Buf$[10;8]
24930    SELECT VAL(Buf$[4;2])
24935    CASE 1
24940       Month$="Jan"
24945    CASE 2
24950       Month$="Feb"
24955    CASE 3
24960       Month$="Mar"
24965    CASE 4
24970       Month$="Apr"
24975    CASE 5
24980       Month$="May"
24985    CASE 6
24990       Month$="Jun"
24995    CASE 7
```

```
25000       Month$="Jul"
25005     CASE 8
25010       Month$="Aug"
25015     CASE 9
25020       Month$="Sep"
25025     CASE 10
25030       Month$="Oct"
25035     CASE 11
25040       Month$="Nov"
25045     CASE 12
25050       Month$="Dec"
25055     END SELECT
25060     SET TIMEDATE DATE(Date$&" "&Month$&" "&Year$)+TIME(Time$)
25065     OFF TIMEOUT
25070     Ret_code=0
25075     SUBEXIT
25080  Timeout:!
25085     OFF TIMEOUT
25090     Ret_code=8
25095  SUBEND
25100  !
25105  SUB Get_date(Date$)
25110  !*************************************************
25115  !*
25120  !*     Get Data on System-clock
25125  !*
25130  !*************************************************
25135     OPTION BASE 1
25140     DIM Buf$[12]
25145     !
25150  Get_date:!
25155     Date$="  -  -  "
25160     Buf$=DATE$(TIMEDATE)
25165     IF LEN(Buf$)=10 THEN
25170       Buf$=" "&Buf$
25175     END IF
25180     Date$[1;2]=Buf$[10;2]
25185     Date$[7;2]=Buf$[1;2]
25190     SELECT Buf$[4;3]
25195     CASE "Jan"
25200       Date$[4;2]=" 1"
25205     CASE "Feb"
25210       Date$[4;2]=" 2"
25215     CASE "Mar"
25220       Date$[4;2]=" 3"
25225     CASE "Apr"
25230       Date$[4;2]=" 4"
25235     CASE "May"
25240       Date$[4;2]=" 5"
25245     CASE "Jun"
25250       Date$[4;2]=" 6"
25255     CASE "Jul"
25260       Date$[4;2]=" 7"
25265     CASE "Aug"
25270       Date$[4;2]=" 8"
25275     CASE "Sep"
25280       Date$[4;2]=" 9"
25285     CASE "Oct"
25290       Date$[4;2]="10"
25295     CASE "Nov"
25300       Date$[4;2]="11"
25305     CASE "Dec"
25310       Date$[4;2]="12"
25315     END SELECT
25320  SUBEND
25325  !
25330  SUB Key_in(Set_no,Inp$,Guide$,INTEGER Key_no)
25335  !*****************************************************
25340  !*
25345  !*    Input Function-key or Data
25350  !*
25355  !*****************************************************
```

```
25360    OPTION BASE 1
25365    DIM Key_buf$[160],Fkey_set$(10)[14]
25370    INTEGER Pointer,Max_point,J,K,Puls
25375 !
25380 !---- Set function-key ------------
25385 !
25390 Key_in:!
25395    RESTORE Function
25400    FOR I=1 TO Set_no
25405       READ Fkey_set$(*)
25410    NEXT I
25415    ON KEY 0 LABEL Fkey_set$(1),3 GOSUB Key0
25420    ON KEY 1 LABEL Fkey_set$(2),3 GOSUB Key1
25425    ON KEY 2 LABEL Fkey_set$(3),3 GOSUB Key2
25430    ON KEY 3 LABEL Fkey_set$(4),3 GOSUB Key3
25435    ON KEY 4 LABEL Fkey_set$(5),3 GOSUB Key4
25440    ON KEY 5 LABEL Fkey_set$(6),3 GOSUB Key5
25445    ON KEY 6 LABEL Fkey_set$(7),3 GOSUB Key6
25450    ON KEY 7 LABEL Fkey_set$(8),3 GOSUB Key7
25455    ON KEY 8 LABEL Fkey_set$(9),3 GOSUB Key8
25460    ON KEY 9 LABEL Fkey_set$(10),3 GOSUB Key9
25465    IF Set_no=9 THEN
25470       ON KEY 19 LABEL Fkey_set$(10),3 GOSUB Key19
25475    END IF
25480 !
25485 !----- Input handling ----------
25490 !
25495    Pointer=0
25500    N=0
25505    Puls=0
25510    IF LEN(Inp$)>0 THEN
25515       Max_point=LEN(Inp$)
25520       OUTPUT 2;CHR$(255)&"#";Guide$;" ";Inp$;
25525       OUTPUT 2;RPT$(CHR$(255)&"<",LEN(Inp$));
25530       ON KNOB .1 GOSUB Knobx
25535       ON KBD GOSUB Keybord
25540    ELSE
25545       ON KBD CALL Dummy
25550       ON KNOB .1 CALL Dummy
25555    END IF
25560    Key_no=-999
25565    WHILE Key_no<-1
25570    END WHILE
25575    OFF KEY
25580    OFF KNOB
25585    OUTPUT 2;CHR$(255)&"#";
25590    CALL Msg_disp(0)
25595    SUBEXIT
25600 !
25605 !----- Keyboard Interrupt ------------
25610 !
25615 Keybord:!
25620    Key_buf$=KBD$
25625    N=1
25630    WHILE N<=LEN(Key_buf$) AND Key_no<-1
25635       IF Key_buf$[N;1]=CHR$(255) THEN
25640          SELECT Key_buf$[N+1;1]
25645          CASE CHR$(255)
25650             N=N+3
25655          CASE "B","<"
25660             IF Pointer>0 THEN
25665                OUTPUT 2;Key_buf$[N;2];
25670                Pointer=Pointer-1
25675             END IF
25680             N=N+2
25685          CASE "E"
25690             Key_no=-1
25695          CASE "J","Y","U"
25700             OUTPUT 2;Key_buf$[N;2];
25705             N=N+2
25710          CASE ">"
25715             IF I<LEN(Inp$) THEN
```

```
25720                OUTPUT 2;Key_buf$[N;2];
25725                Pointer=Pointer+1
25730             END IF
25735             N=N+2
25740          CASE ELSE
25745             N=N+2
25750          END SELECT
25755       ELSE
25760          IF Pointer>=Max_point THEN
25765             N=999
25770          ELSE
25775             OUTPUT 2;Key_buf$[N;1];
25780             Pointer=Pointer+1
25785             Inp$[Pointer;1]=Key_buf$[N;1]
25790             N=N+1
25795          END IF
25800       END IF
25805    END WHILE
25810    RETURN
25815 !
25820 !---- Function-key Interrupt ------
25825 !
25830 Key0:!
25835    Key_no=0
25840    RETURN
25845 Key1:!
25850    Key_no=1
25855    RETURN
25860 Key2:!
25865    Key_no=2
25870    RETURN
25875 Key3:!
25880    Key_no=3
25885    RETURN
25890 Key4:!
25895    Key_no=4
25900    RETURN
25905 Key5:!
25910    Key_no=5
25915    RETURN
25920 Key6:!
25925    Key_no=6
25930    RETURN
25935 Key7:!
25940    Key_no=7
25945    RETURN
25950 Key8:!
25955    Key_no=8
25960    RETURN
25965 Key9:!
25970    Key_no=9
25975    RETURN
25980 Key19:!
25985    Key_no=19
25990    RETURN
25995 !
26000 !------ Knob Interrupt -----------
26005 !
26010 Knobx:!
26015    Puls=Puls+KNOBX
26020    WHILE ABS(Puls)>5
26025       IF Puls>0 THEN
26030          Puls=Puls-6
26035          IF Pointer<LEN(Inp$) THEN
26040             OUTPUT 2;CHR$(255)&">";
26045             Pointer=Pointer+1
26050          END IF
26055       ELSE
26060          Puls=Puls+6
26065          IF Pointer>0 THEN
26070             OUTPUT 2;CHR$(255)&"<";
26075             Pointer=Pointer-1
```

```
26080          END IF
26085       END IF
26090    END WHILE
26095    RETURN
26100 !
26105 !------ Function Label -------------
26110 !
26115 Function:!
26120    DATA " Input data    "," Measurement  "," Print result " ! 1
26125    DATA "               ","              "
26130    DATA "               ","              ","               "
26135    DATA "               ","              "
26140    DATA " A solution    "," B solution   ","    Vial       " ! 2
26145    DATA "  Print        ","              "
26150    DATA "               ","              ","               "
26155    DATA "               ","   Menu       "
26160    DATA "previous data  "," 1st sampling ","  2nd sampling " ! 3
26165    DATA "    Next       ","              "
26170    DATA "               ","              ","               "
26175    DATA "               ","   Return     "
26180    DATA "               ","              ","               " ! 4
26185    DATA "    Back       ","              "
26190    DATA "               ","              ","               "
26195    DATA "               ","   Return     "
26200    DATA "               ","              ","               " ! 5
26205    DATA "               ","              "
26210    DATA "               ","              ","               "
26215    DATA "               ","   Return     "
26220    DATA "   Start       "," Set-up (DL20)","               " ! 6
26225    DATA "               ","              "
26230    DATA "               ","              ","               "
26235    DATA "               ","    Menu      "
26240  ! DATA "  Directory    ","  Display     ","  Print        " ! 7
26245    DATA "  Directory    ","              ","  Print        " ! 7
26250    DATA " Format disk   ","              "
26255    DATA "               ","              ","               "
26260    DATA "               ","    Menu      "
26265    DATA "   Next        ","              ","               " ! 8
26270    DATA "               ","              "
26275    DATA "               ","              ","               "
26280    DATA "               ","   Return     "
26285    DATA "               ","              ","               " ! 9
26290    DATA "               ","              "
26295    DATA "               ","              ","               "
26300    DATA "               ","    (END)     "
26305 SUBEND
26310 !
26315 SUB Msg_disp(INTEGER Msg_no)
26320 !*****************************************************
26325 !*
26330 !*    Display Message
26335 !*
26340 !*****************************************************
26345    OPTION BASE 1
26350    COM /Message/ INTEGER Msg_flag
26355    DIM Buf$[32]
26360    INTEGER I
26365    !
26370 Msg_disp:!
26375    IF Msg_no>0 AND Msg_no<19 THEN
26380       RESTORE Message
26385       FOR I=1 TO Msg_no
26390          READ Buf$
26395       NEXT I
26400       BEEP
26405      !CALL K_move(0,21)
26410      !CALL K_label(Buf$)
26415       CSIZE 8,.4
26420       MOVE 5,58
26425       LABEL Buf$
26430       Msg_flag=Msg_no
26435    ELSE
```

```
26440       IF Msg_flag<>0 THEN
26445          RESTORE Message
26450          FOR I=1 TO Msg_flag
26455             READ Buf$
26460          NEXT I
26465         !Buf$=RPT$("!!",16)
26470         !CALL K_move(0,21)
26475         !CALL K_label(Buf$)
26480          CSIZE 8,.4
26485          MOVE 5,58
26490          PEN -1
26495          LABEL Buf$
26500          PEN 1
26505          Msg_flag=0
26510       END IF
26515    END IF
26520    SUBEXIT
26525 !
26530 !----- Message data --------
26535 !
26540 Message:!
26545    DATA "Error in data input                "
26550    DATA "Error in input format              "
26555    DATA "No report number                   "
26560    DATA "Error in data preset               "
26565    DATA "Wrong report number                "
26570    DATA "Insutticient number of vial        "
26575    DATA "Insutticient number of cups        "
26580    DATA "Malfunction                        "
26585    DATA "Abort Measurement                  "
26590    DATA "Wrong disk                         "
26595    DATA "Disk overflow                      "
26600    DATA "Disk set error                     "
26605    DATA "Disk I/O error                     "
26610    DATA "                                   "
26615    DATA "In set-up (DL-20)                  "
26620    DATA "End of measurement                 "
26625    DATA "Set Automatic-Mode !               "
26630    DATA "                                   "
26635 SUBEND
26640 !
26645 SUB Kanji_conv(Buf$,Kanji$)
26650 !*****************************************************
26655 !*
26660 !*     Conversion ASCII to Kanji-Code (half-mode)
26665 !*
26670 !*****************************************************
26675    INTEGER I
26680    !
26685 Kanji_conv:!
26690    Kanji$=""
26695    FOR I=1 TO LEN(Buf$)
26700       SELECT Buf$[I;1]
26705       CASE "!" TO "~"
26710          Kanji$[2*I-1;2]=CHR$(46)&Buf$[I;1]
26715       CASE "｡" TO "ﾟ"
26720          Kanji$[2*I-1;2]=CHR$(47)&CHR$(NUM(Buf$[I;1])-128)
26725       CASE ELSE
26730          Kanji$[2*I-1;2]=CHR$(47)&CHR$(96)
26735       END SELECT
26740    NEXT I
26745 SUBEND
26750 !
26755 SUB Dummy
26760 !*******************************************
26765 !*
26770 !*      Dummy Routine
26775 !*
26780 !*******************************************
26785 SUBEND
26790 !
26795 SUB Test10
```

```
26800 !***************
26805 !*
26810 !***************
26815    COM /Robot/ INTEGER Now_posi,Hand,REAL Length,Hight,Turn
26820    INTEGER Ret_code
26825    REAL H
26830 Test10:!
26835    INPUT "Hight=",H
26840    IF H=-1 THEN GOTO Test12
26845    IF H=0 THEN
26850       CALL Motor_1(3,-1,Ret_code)
26855       Hight=0
26860    ELSE
26865       CALL Motor_1(2,H,Ret_code)
26870       Hight=Hight+H
26875    END IF
26880    PRINT TABXY(10,10),Hight
26885    GOTO Test10
26890 Test12:!
26895    INPUT "Turn =",H
26900    IF H=-1 THEN GOTO Test13
26905    IF H=0 THEN
26910       CALL Motor_2(4,-1,Ret_code)
26915       Turn=0
26920    ELSE
26925       CALL Motor_2(2,H,Ret_code)
26930       Turn=Turn+H
26935    END IF
26940    PRINT TABXY(10,11),Turn
26945    GOTO Test12
26950 Test13:!
26955    INPUT "Length=",H
26960    IF H=-1 THEN GOTO Test14
26965    IF H=0 THEN
26970       CALL Motor_3(4,-1,Ret_code)
26975       Length=0
26980    ELSE
26985       CALL Motor_3(2,H,Ret_code)
26990       Length=Length+H
26995    END IF
27000    PRINT TABXY(10,12),Length
27005    GOTO Test13
27010 Test14:!
27015    INPUT "Hand=",H
27020    IF H=-1 THEN GOTO Test10
27025    IF H=0 THEN
27030       CALL Motor_4(3,-1,Ret_code)
27035       Hand=0
27040    ELSE
27045       CALL Motor_4(1,H,Ret_code)
27050       Hand=1
27055    END IF
27060    PRINT TABXY(10,13),Hand
27065    GOTO Test14
27070 SUBEND
27075 !
27080 SUB Test20
27085 !*****************
27090 !*
27095 !*****************
27100    COM /Robot/ INTEGER Now_posi,Hand,REAL Length,Hight,Turn
27105    INTEGER I,J,Ret_code
27110 Test20:!
27115    INPUT "(arm)POSI,CUP",I,J
27120    CALL Move_arm(I,J,Ret_code)
27125    PRINT Length,Hight,Turn,Ret_code
27130    INPUT "(hand) POSI,CUP",I,J
27135    IF I>=0 THEN
27140       CALL Move_hand(I,J,Ret_code)
27145       PRINT Length,Hight,Turn,Ret_code
27150    END IF
27155    GOTO Test20
27160 SUBEND
27165 !
```

List of movement test program "MOVE TEST"

```
1000  !    RE-STORE "MOVE_TEST"
1010  !   ************************************************
1020  !   *    The Automated Gravimetric Sampling System   *
1030  !   *                                        (PNC)   *
1040  !   *              60B-010L                          *
1050  !   ************************************************
1060  OPTION BASE 0
1070   DIM Label$(4,1)[15]
1080   OUTPUT 2;"SCRATCH KEYX";
1090   OUTPUT 2;"KX";
1100   GCLEAR
1110   PRINTER IS 30
1120     PRINT CHR$(27)&"B"&CHR$(72);
1130   PRINTER IS CRT
1140   DISP "GMACS Clear"
1150   Io_gmacs=709
1160   Pm_gmacs=710
1170   Rs_232=9
1180   Cw=0
1190   Ccw=1
1200   RESTORE Key_label
1210   READ Label$(*)
1220 Key_label:    DATA Right,Left,Down,Up,Back,Forward,Off,Catch,Sample in,Samp
le out
1230        !
1240   CONTROL Rs_232,3;2400              ! 2400 baud
1250   CONTROL Rs_232,4;2+0+8+16          ! 7-bit ASCII,1 stop bit,even parity
1260   CALL Init_22(Io_gmacs,@Gmacs_io)   !  GMACS IO CLEAR
1270   CALL Init_22(Pm_gmacs,@Gmacs_pm)   !  GMACS PM CLEAR
1280    FOR Channel=0 TO 4
1290        CALL Mi_pm(@Gmacs_pm,Channel,0,2,50,600,1000)
1300    NEXT Channel
1310   CALL Init_dl20(Rs_232,@Rs_232c)    !  DL 20 RESET
1320 Key_set1:  OFF KEY
1330   OUTPUT 2;"#X";
1340   DISP "";
1350   ON KEY 0 LABEL "Cup turn      " GOTO Cup_turn
1360   ON KEY 1 LABEL "Mixing ON     " GOTO Stirrer_on
1370   ON KEY 2 LABEL "Vial turn     " GOTO Vial_turn
1380   ON KEY 3 LABEL "Bal.cover op." GOTO Cover_open
1390   ON KEY 4 LABEL "END PRG.      " GOTO End_prg
1400   ON KEY 5 LABEL "              " GOTO Key_set1
1410   ON KEY 6 LABEL "Mixing OFF    " GOTO Stirrer_off
1420   ON KEY 7 LABEL "              " GOTO Key_set1
1430   ON KEY 8 LABEL "Bal.cover cl." GOTO Cover_shut
1440   ON KEY 9 LABEL "EXIT          " GOTO Key_set2
1450   LOOP
1460     CALL Status_disp(@Gmacs_io,@Gmacs_pm)
1470   END LOOP
1480 Key_set2:  OFF KEY
1490   DISP "";
1500   OUTPUT 2;"#X";
1510   ON KEY 0 LABEL "ROBOT Turn    " GOTO Turn
1520   ON KEY 1 LABEL "ROBOT Up/Down" GOTO Up_dn
1530   ON KEY 2 LABEL "ROBOT Arm     " GOTO Arm
1540   ON KEY 3 LABEL "ROBOT Catch   " GOTO Cach
1550   ON KEY 4 LABEL "Sampling      " GOTO Sample
1560   ON KEY 5 LABEL "Bal. Data     " GOTO Balance_data
1570   ON KEY 6 LABEL "Bal. Tare     " GOTO Tear_reset
1580   ON KEY 7 LABEL "DL-20 Start   " GOTO Dl_20
1590   ON KEY 8 LABEL "Bal.Data PRT  " GOTO Bal_print
1600   ON KEY 9 LABEL "EXIT          " GOTO Key_set1
1610   LOOP
1620     CALL Status_disp(@Gmacs_io,@Gmacs_pm)
1630   END LOOP
1640        !
1650        !
1660  !!!!!!!!!!!    KEY SET 1   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
1670 Cup_turn:DISP "******** Rotate the turntable for the cups ***********"
1680   OFF KEY
1690   CALL Cup_turn(@Gmacs_io)
1700   GOTO Key_set1
1710        !
1720 Stirrer_on:DISP " ***************  Rotate the stirrer ******************"
1730   OFF KEY
1740   CALL Sb_22(@Gmacs_io,0,2)
1750   GOTO Key_set1
1760        !
1770 Vial_turn:DISP "******** Rotate the turntable for the viales ********"
1780   OFF KEY
1790   CALL Vial_turn(@Gmacs_io)
1800   GOTO Key_set1
1810        !
1820 Cover_open:DISP " ************ Open the balance cover ******************"
1830   OFF KEY
1840   CALL Cover_open(@Gmacs_io)
1850   GOTO Key_set1
1860        !
1870 Stirrer_off:DISP " *********** Stop the stirrer ************************"
1880   OFF KEY
1890   CALL Rb_22(@Gmacs_io,0,2)
1900   GOTO Key_set1
1910        !
1920 Cover_shut:DISP " *********** Close the balance cover ****************"
1930   OFF KEY
1940   CALL Cover_shut(@Gmacs_io)
1950   GOTO Key_set1
1960        !
1970  !!!!!!!!!!    KEY SET 2   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
1980 Turn:       DISP "*****************   ROBOT TURN    *****************"
1990             Ch=0
2000             Pulse=5000
2010             GOTO Select
2020 Up_dn:      DISP "***************    ROBOT UP/DOWN    ***************"
2030             Ch=1
2040             Pulse=35000
2050             GOTO Select
2060 Arm:        DISP "******  ROBOT Extension and Contraction  *******"
2070             Ch=2
2080             Pulse=1500
2090             GOTO Select
2100 Cach:       DISP "*************     ROBOT CATCH      **************"
2110             Ch=3
2120             Pulse=400
2130             GOTO Select
2140 Sample: !
2150             Ch=4
2160             Pulse=7000
2170             INPUT "Sample Pulse",Pulse
2180             DISP "*********** Sampling Mechanism *********"
2190             GOTO Select
2200             !
2210 Select:!
2220   OFF KEY
2230   ON KEY 0 LABEL "Reset  " GOTO Position
2240   ON KEY 1 LABEL Label$(Ch,0) GOTO Pm_cw
2250   ON KEY 2 LABEL Label$(Ch,1) GOTO Pm_ccw
2260   ON KEY 5 LABEL "Stop  " GOTO Pm_stop
2270   ON KEY 9 LABEL "EXIT   " GOTO Exit
2280   LOOP
2290     CALL Status_disp(@Gmacs_io,@Gmacs_pm)
2300   END LOOP
2310 Position:!
2320   SELECT Ch
2330   CASE 0,4                          ! CCW
2340     Cw_ccw=1
2350   CASE ELSE
2360      Cw_ccw=0                       ! CW
2370   END SELECT
2380   CALL Mh_pm(@Gmacs_pm,Ch,Cw_ccw)
```

```
2390    GOSUB Limit_st
2400    GOTO Select
2410 Pm_cw:!
2420    CALL Ma_pm(@Gmacs_pm,Ch,Cw,Pulse)
2430    GOSUB Limit_st
2440    GOTO Select
2450 Pm_ccw:!
2460    CALL Ma_pm(@Gmacs_pm,Ch,Ccw,Pulse)
2470    GOSUB Limit_st
2480    GOTO Select
2490 Pm_stop:!
2500    CALL Ms_pm(@Gmacs_pm,Ch,0)
2510    GOTO Select
2520 Exit:!
2530    CALL Ms_pm(@Gmacs_pm,Ch,0)
2540    GOTO Key_set2
2550 Limit_st:!
2560      LOOP
2570        CALL Ds_pm(@Gmacs_pm,Ch,2,St)
2580      EXIT IF BIT(St,5)=0                                  !  PM Stop
2590      END LOOP
2600      RETURN
2610         !
2620 Balance_data:DISP " ************  BAL. Data INPUT  ********************"
2630    OFF KEY
2640    ON KEY 9 LABEL "EXIT" GOTO Key_set2
2650 Bal_disp:!
2660    LOOP
2670      CALL Balance_data(@Gmacs_io,Balance_data$,Bl_data)
2680      DISP "Balance Data ";Balance_data$[1,2]&"   ";Bl_data;" g              "
2690    END LOOP
2700 Tear_reset:DISP "****************   TARE RESET   ******************"
2710    OUTPUT @Gmacs_io;"TA"
2720    WAIT 2
2730    GOTO Key_set2
2740         !
2750 DL_20: !****************** DL-20 Start ****************************
2760    OFF KEY
2770    Kishaku=20
2780    INPUT "Dilution amount Input  [ml] ",Kishaku
2790    DISP "****************** DL-20 Start ****************************"
2800    CALL DL_20(@Rs_232c,Kishaku)
2810    GOTO Key_set2
2820      !
2830 Bal_print:!
2840    CALL Bal_print(@Gmacs_io)
2850    GOTO Key_set2
2860    !
2870 End_prg:!
2880          PRINT CHR$(12)
2890    END
2900      !
2910      !
2920    SUB Status_disp(@Gmacs_io,@Gmacs_pm)
2930     ! ********************* LIMIT PRINT ************************************
2940      OPTION BASE 0
2950      CALL Di_22(@Gmacs_io,0,Bit0,Bit0$)
2960      CALL Di_22(@Gmacs_io,1,Bit1,Bit1$)
2970      CALL Di_22(@Gmacs_io,2,Bit2,Bit2$)
2980      PRINT TABXY(1,1);
2990      PRINT USING "10X,K,/";"***************  Status(1:ON,0:OFF) ************
****"
3000      PRINT USING 3010;"Auto ",BIT(Bit0,0),"Manu.",BIT(Bit0,1),"Stop",BIT(Bit
0,2)
3010      IMAGE 10X,2(5A,3X,Z,15X),5A,3X,Z
3020      PRINT USING "10X,70A ";RPT$("-",67)
3030 Io_im:IMAGE 10X,25A,Z,10X,25A,Z
3040      PRINT USING Io_im;"Cup reset         ",BIT(Bit0,3),"Vial reset     ",BI
T(Bit1,0)
3050      PRINT USING Io_im;"Cup position        ",BIT(Bit0,4),"Vial position
",BIT(Bit1,1)
3060      PRINT USING Io_im;"Cup hold posi.       ",BIT(Bit0,5),"Vial hold posi.
```

```
Down ",BIT(Bit1,2)
3070      PRINT USING "46X,25A,Z";"Vial hold posi. up   ",BIT(Bit1,3)
3080      PRINT USING "10X,70A ";RPT$("-",67)
3090      PRINT USING "10X,25A,Z";"Bal.cover Open ",BIT(Bit1,4)
3100      PRINT USING "10X,25A,Z";"Bal.cover Close",BIT(Bit1,5)
3110      PRINT USING "10X,70A ";RPT$("-",67)
3120      FOR I=0 TO 4
3130        CALL Ds_pm(@Gmacs_pm,I,2,St(I))
3140      NEXT I
3150      PRINT USING "10X,19A,2A,5X,3A,27X,2A,5X,3A";"Motor Limit    ","CW","CCW"
,"CW","CCW"
3160 Pm_im1:IMAGE #,10X,19A,X,Z,7X,Z
3170 Pm_im2:IMAGE 8X,19A,X,Z,7X,Z
3180      PRINT USING Pm_im1;"Turn       ",BIT(St(0),3),BIT(St(0),2)
3190      PRINT USING Pm_im2;"Up/Down    ",BIT(St(1),3),BIT(St(1),2)
3200      PRINT USING Pm_im1;"Arm        ",BIT(St(2),3),BIT(St(2),2)
3210      PRINT USING Pm_im2;"Catch      ",BIT(St(3),3),BIT(St(3),2)
3220      PRINT USING Pm_im1;"Sampling   ",BIT(St(4),3),BIT(St(4),2)
3230    SUBEND
3240       !
3250       !
3260    SUB Cup_turn(@Gmacs_io)
3270    ! ************ Rotate the turntable for the cups ********
3280      ON KEY 9 LABEL " EXIT" GOTO Stop
3290      CALL Sb_22(@Gmacs_io,0,1)
3300      LOOP
3310        CALL Di_22(@Gmacs_io,0,Bit0,Bit0$)
3320      EXIT IF BIT(Bit0,4)=0                          !   ----- POSITION OFF
3330      END LOOP
3340      WAIT .1
3350      CALL Rb_22(@Gmacs_io,0,1)
3360      LOOP
3370        CALL Di_22(@Gmacs_io,0,Bit0,Bit0$)
3380      EXIT IF BIT(Bit0,4)=1                          !   ----- POSITION ON
3390      END LOOP
3400      SUBEXIT
3410 Stop:CALL Rb_22(@Gmacs_io,0,1)
3420    SUBEND
3430       !
3440       !
3450    SUB Vial_turn(@Gmacs_io)
3460    !************ Rotate the turntable for the vials *********
3470      ON KEY 9 LABEL " EXIT" GOTO Stop
3480      CALL Sb_22(@Gmacs_io,0,4)
3490      LOOP
3500        CALL Di_22(@Gmacs_io,1,Bit1,Bit1$)
3510      EXIT IF BIT(Bit1,1)=0                          !   ----- POSITION OFF
3520      END LOOP
3530      WAIT .1
3540      CALL Rb_22(@Gmacs_io,0,4)
3550      LOOP
3560        CALL Di_22(@Gmacs_io,1,Bit1,Bit1$)
3570      EXIT IF BIT(Bit1,1)=1                          !   ----- POSITION ON
3580      END LOOP
3590      SUBEXIT
3600 Stop:CALL Rb_22(@Gmacs_io,0,4)
3610    SUBEND
3620       !
3630       !
3640    SUB Cover_open(@Gmacs_io)
3650    ! ********************** Open the balance cover ************************
3660      ON KEY 9 LABEL " EXIT" GOTO Stop
3670      CALL Sb_22(@Gmacs_io,0,8)
3680      LOOP
3690        CALL Di_22(@Gmacs_io,1,Bit1,Bit1$)
3700      EXIT IF BIT(Bit1,4)=1                          !   ----- Cover Open
3710      END LOOP
3720 Stop:CALL Rb_22(@Gmacs_io,0,8)
3730    SUBEND
3740       !
3750       !
3760    SUB Bal_print(@Gmacs_io)
```

```
3770   ! ********************** Bal. Data Print **********************
3780   OPTION BASE 1
3790   DIM Bal_data(50)
3800   MAT Bal_data= (0)
3810   Count=1
3820   OUTPUT @Gmacs_io;"TD"
3830   ENTER @Gmacs_io;Td$              !  --------- DATE TIME
3840   OFF KEY
3850   DISP
3860 Key_set:!
3870   ON KEY 0 LABEL "BAL.DATA" GOTO Bal_print
3880   ON KEY 1 LABEL "BAL.TEAR       " GOTO Bal_tear
3890   ON KEY 2 LABEL "BAL.COVER op." GOTO Cover_open
3900   ON KEY 3 LABEL "BAL.COVER cl." GOTO Cover_shut
3910   ON KEY 9 LABEL "EXIT           " GOTO Exit
3920   LOOP
3930   END LOOP
3940 Bal_print:  !
3950   OFF KEY
3960   Bal_data(Count)=0
3970   FOR I=1 TO 10
3980     CALL Balance_data(@Gmacs_io,Balance_data$,Bl_data)
3990       Bl_data=INT(Bl_data*10000)/10000
4000     IF Balance_data$[1,2]="S " THEN
4010        Bal_data(Count)=Bl_data+Bal_data(Count)
4020     ELSE
4030        GOTO Bal_print
4040     END IF
4050    NEXT I
4060    Bal_data(Count)=Bal_data(Count)/10
4070    DISP USING "2D,5X,20A,M3D.4D,2X,A";Count,"Balance Data",Bal_data(Count),
"g"
4080    PRINTER IS 30
4090    IF Count=1 THEN
4100       PRINT USING "5/"
4110       PRINT USING "30X,K,3/";"Balance Data Result"
4120       PRINT USING "41X,K,2/";"Signature NSB"&RPT$("_",20)
4130       PRINT USING "50X,K,2/";"IAEA"&RPT$("_",20)
4140       PRINT USING "50X,K,2/";"Date  19"&Td$[1,8]
4150    END IF
4160    PRINT USING 4170;Count,"Balance Data",Bal_data(Count),"g"
4170    IMAGE  10X,"(",2D,")",5X,20A,M3D.4D,2X,A
4180    PRINTER IS CRT
4190    Count=Count+1
4200    IF Count=11 THEN Average
4210    GOTO Bal_print
4220      !
4230 Bal_tear:  !
4240    OFF KEY
4250    OUTPUT @Gmacs_io;"TA"
4260    WAIT 2
4270    GOTO Key_set
4280      !
4290 Cover_open:  !
4300   OFF KEY
4310   CALL Cover_open(@Gmacs_io)
4320   GOTO Key_set
4330   !
4340 Cover_shut:!
4350   OFF KEY
4360   CALL Cover_shut(@Gmacs_io)
4370   GOTO Key_set
4380   !
4390 Average:  !
4400   B=0
4410   C=0
4420   FOR I=1 TO Count-1
4430      B=B+Bal_data(I)
4440      C=C+Bal_data(I)^2
4450   NEXT I
4460    X=B/(Count-1)
4470    Sigma=SQR((C-(Count-1)*X^2)/((Count-1)-1))
```

```
4480    Cv=Sigma/X*100
4490    DISP USING "20A,M3D.4D,2X,A,5X,5A,M3D.4D,A";"AVERAGE ",X,"g","CV   ",Cv,"
%"
4500   PRINTER IS 30
4510   PRINT USING "10X,K";RPT$("-",40)
4520 Im_1:  IMAGE 19X,20A,M3D.4D,2X,A
4530    PRINT USING Im_1;"Average",X,"g"
4540    PRINT USING Im_1;"Sigma",Sigma,"g"
4550    PRINT USING Im_1;"CV (%)",Cv,"%"
4560    PRINT CHR$(12)
4570   PRINTER IS CRT
4580   Count=1
4590   MAT Bal_data= (0)
4600   GOTO Key_set
4610 Exit:  !
4620   PRINTER IS CRT
4630   SUBEND
4640      !
4650      !
4660   SUB Dum2
4670      !
4680      !
4690   SUBEND
4700      !
4710      !
4720   SUB Cover_shut(@Gmacs_io)
4730   ! ************************ Close the balance cover ********************
4740     ON KEY 9 LABEL " EXIT" GOTO Stop
4750     CALL Sb_22(@Gmacs_io,0,16)
4760     LOOP
4770       CALL Di_22(@Gmacs_io,1,Bit1,Bit1$)
4780     EXIT IF BIT(Bit1,5)=1                          !   ----- ｶﾊﾞｰ ﾍｲ ON
4790     END LOOP
4800 Stop:CALL Rb_22(@Gmacs_io,0,16)
4810   SUBEND
4820       !
4830       !
4840   SUB Balance_data(@Gmacs_io,Balance_data$,Bl_data)
4850    ! ********************  Bal. Data Input  ********************
4860     ON TIMEOUT SC(@Gmacs_io),5 GOTO Err
4870     ON INTR SC(@Gmacs_io) GOTO Err
4880     OUTPUT @Gmacs_io;"BD"
4890     ENTER @Gmacs_io;Balance_data$
4900     IF LEN(Balance_data$)>5 THEN
4910        Bl_data=VAL(Balance_data$[5,12])/10.
4920     ELSE
4930        GOTO 4880
4940     END IF
4950     SUBEXIT
4960 Err:   CALL Err_check(@Gmacs_io,"BD")
4970   SUBEND
4980    !
4990    !
5000   SUB Dl_20(@Rs_232c,Kishaku)
5010    ! ************************** DL 20 Start ***************************
5020    ON TIMEOUT SC(@Rs_232c),2 GOTO Exit_sub
5030    Buret=Kishaku/20                          !    20 ml buret
5040    OUTPUT @Rs_232c;VAL$(Buret)               !    buretter strokes number
5050    OUTPUT @Rs_232c;"M"                       !    BURET key
5060    WAIT Buret*30
5070    SUBEXIT
5080    !
5090 Exit_sub:!
5100          DISP "DL 20 TIMEOUT"
5110          BEEP 1000,3
5120   SUBEND
5130   SUB Init_dl20(Rs_232,@Rs_232c)
5140   ! ************************** DL 20 RESTE ***************************
5150    ON TIMEOUT Rs_232,2 GOTO Exit_sub
5160    ASSIGN @Rs_232c TO Rs_232
5170    OUTPUT @Rs_232c;"L"
5180    SUBEXIT
```

```
5190 Exit_sub:!
5200          DISP "DL 20 TIMEOUT"
5210          BEEP 1000,3
5220    SUBEND
5230       SUB Init_22(Gmacs,@Gmacs)
5240  ! ************************************************************************
5250  ! *****     7522  Sub  <<  Initialize  >>                          *****
5260  ! ************************************************************************
5270          ASSIGN @Gmacs TO Gmacs
5280          ON TIMEOUT SC(@Gmacs),10 CALL Time_out
5290          S=SPOLL(@Gmacs)
5300          CLEAR @Gmacs
5310          WAIT 5
5320       SUBEND
5330       SUB Err_check(@Gmacs,Cm$)
5340  ! ************************************************************************
5350  ! *****     7522  Sub  <<  Error  Check  >>                        *****
5360  ! ************************************************************************
5370          ON TIMEOUT SC(@Gmacs),5 CALL Time_out
5380          S=SPOLL(@Gmacs)
5390          IF BIT(S,6) AND (BIT(S,1) OR BIT(S,5)) THEN
5400               CALL Err_22(Cm$)
5410          ELSE
5420               CALL Time_out
5430          END IF
5440       SUBEND
5450       SUB Time_out
5460  ! ************************************************************************
5470  ! *****     7522  Sub  <<  Time  out  >>                           *****
5480  ! ************************************************************************
5490          OFF TIMEOUT
5500          PRINT CHR$(12)
5510          PRINT "7522  (GMACS)  TIME OUT ERROR !"
5520          STOP
5530       SUBEND
5540       SUB Err_22(Cm$)
5550  ! ************************************************************************
5560  ! *****     7522  Sub  <<  Error  Massage  Display  >>             *****
5570  ! ************************************************************************
5580          PRINT CHR$(12)
5590          PRINT "*    GMACS  ERROR    ";Cm$;"    COMAND"
5600          PRINT
5610          PRINT "     ERROR  23 SYSTEM ERROR "
5620          PRINT "     ERROR  88 COMMAND ERROR "
5630          PRINT "     ERROR  89 PARAMETER ERROR "
5640          PRINT "     ERROR  131 GPIB TIME OUT ERROR "
5650          STOP
5660       SUBEND
5670       SUB Di_22(@Gmacs,Channel,Di_data,Binary$)
5680  ! ************************************************************************
5690  ! *****     7522  Sub  <<  DI_22  >>                               *****
5700  ! ************************************************************************
5710          ON TIMEOUT SC(@Gmacs),5 GOTO Err
5720          ON INTR SC(@Gmacs) GOTO Err
5730          ENABLE INTR SC(@Gmacs);2
5740          OUTPUT @Gmacs;"ST,"&VAL$(IVAL("01101100",2))
5750          OUTPUT @Gmacs;"DI,"&VAL$(Channel)
5760          ENTER @Gmacs;Di_data
5770          OFF TIMEOUT
5780          OFF INTR
5790          Bin$=IVAL$(Di_data,2)
5800          Binary$=Bin$[9]
5810          SUBEXIT
5820 Err:     CALL Err_check(@Gmacs,"DI")
5830       SUBEND
5840       SUB Dr_22(@Gmacs)
5850  ! ************************************************************************
5860  ! *****     7522  Sub  <<  DR_22  >>                               *****
5870  ! ************************************************************************
5880          ON TIMEOUT SC(@Gmacs),5 CALL Time_out
5890          OUTPUT @Gmacs;"DR"
5900          OFF TIMEOUT
```

```
5910        SUBEND
5920        SUB Sb_22(@Gmacs,Channel,On_bit)
5930   ! ************************************************************************
5940   ! *****      7522  Sub  <<  SB_22  >>                              *****
5950   ! ************************************************************************
5960              ON TIMEOUT SC(@Gmacs),5 GOTO Err
5970              ON INTR SC(@Gmacs) GOTO Err
5980              ENABLE INTR SC(@Gmacs);2
5990              OUTPUT @Gmacs;"ST,"&VAL$(IVAL("01101100",2))
6000              OUTPUT @Gmacs;"SB,"&VAL$(Channel)&","&VAL$(On_bit)
6010              OFF TIMEOUT
6020              OFF INTR
6030              SUBEXIT
6040 Err:         CALL Err_check(@Gmacs,"SB")
6050        SUBEND
6060        SUB Rb_22(@Gmacs,Channel,Off_bit)
6070   ! ************************************************************************
6080   ! *****      7522  Sub  <<  RB_22  >>                              *****
6090   ! ************************************************************************
6100              ON TIMEOUT SC(@Gmacs),5 GOTO Err
6110              ON INTR SC(@Gmacs) GOTO Err
6120              ENABLE INTR SC(@Gmacs);2
6130              OUTPUT @Gmacs;"ST,"&VAL$(IVAL("01101100",2))
6140              OUTPUT @Gmacs;"RB,"&VAL$(Channel)&","&VAL$(Off_bit)
6150              OFF TIMEOUT
6160              OFF INTR
6170              SUBEXIT
6180 Err:         CALL Err_check(@Gmacs,"RB")
6190        SUBEND
6200        SUB Mi_pm(@Gmacs,Channel,Raji,Mot_kind,Kagen_rate,Kosok_rate,Pls_coun
t)
6210   ! ************************************************************************
6220   ! *****      7522  Pulse  Motor  Sub  <<  MI_PM  >>                *****
6230   ! ************************************************************************
6240              ON TIMEOUT SC(@Gmacs),5 GOTO Err
6250              ON INTR SC(@Gmacs) GOTO Err
6260              ENABLE INTR SC(@Gmacs);2
6270              OUTPUT @Gmacs;"ST,"&VAL$(IVAL("01101100",2))
6280              OUTPUT @Gmacs;"MI,"&VAL$(Channel)&","&VAL$(Raji)&","&VAL$(Mot_ki
nd)&","&VAL$(Kagen_rate)&","&VAL$(Kosok_rate)&","&VAL$(Pls_count)
6290              OFF TIMEOUT
6300              OFF INTR
6310              SUBEXIT
6320 Err:         CALL Err_check(@Gmacs,"MI")
6330        SUBEND
6340        SUB Ds_pm(@Gmacs,Channel,Status_no,Status)
6350   ! ************************************************************************
6360   ! *****      7522  Pulse  Motor  Sub  <<  DS_PM  >>                *****
6370   ! ************************************************************************
6380              ON TIMEOUT SC(@Gmacs),1 GOTO Err
6390              ON INTR SC(@Gmacs) GOTO Err
6400              ENABLE INTR SC(@Gmacs);2
6410              OUTPUT @Gmacs;"ST,"&VAL$(IVAL("01101100",2))
6420              OUTPUT @Gmacs;"DS,"&VAL$(Channel)&","&VAL$(Status_no)&","&VAL$(S
tatus)
6430              ENTER @Gmacs;Status
6440              OFF TIMEOUT
6450              OFF INTR
6460              SUBEXIT
6470 Err:         CALL Err_check(@Gmacs,"DS")
6480        SUBEND
6490        SUB Ma_pm(@Gmacs,Channel,Revolution,Pls_count)
6500   ! ************************************************************************
6510   ! *****      7522  Pulse  Motor  Sub  <<  MA_PM  >>                *****
6520   ! ************************************************************************
6530              ON TIMEOUT SC(@Gmacs),5 GOTO Err
6540              ON INTR SC(@Gmacs) GOTO Err
6550              ENABLE INTR SC(@Gmacs);2
6560              OUTPUT @Gmacs;"ST,"&VAL$(IVAL("01101100",2))
6570              OUTPUT @Gmacs;"MA,"&VAL$(Channel)&","&VAL$(Revolution)&","&VAL$(
Pls_count)
6580              OFF TIMEOUT
```

```
6590            OFF INTR
6600            SUBEXIT
6610 Err:       CALL Err_check(@Gmacs,"MA")
6620       SUBEND
6630       SUB Mh_pm(@Gmacs,Channel,Revolution)
6640  ! ********************************************************************************
6650  ! *****     7522  Pulse  Motor  Sub  <<  MH_PM  >>                      *****
6660  ! ********************************************************************************
6670            ON TIMEOUT SC(@Gmacs),5 GOTO Err
6680            ON INTR SC(@Gmacs) GOTO Err
6690            ENABLE INTR SC(@Gmacs);2
6700            OUTPUT @Gmacs;"ST,"&VAL$(IVAL("01101100",2))
6710            OUTPUT @Gmacs;"MH,"&VAL$(Channel)&","&VAL$(Revolution)
6720            OFF TIMEOUT
6730            OFF INTR
6740            SUBEXIT
6750 Err:       CALL Err_check(@Gmacs,"MH")
6760       SUBEND
6770       SUB Ms_pm(@Gmacs,Channel,Stop)
6780  ! ********************************************************************************
6790  ! *****     7522  Pulse  Motor  Sub  <<  MS_PM  >>                      *****
6800  ! ********************************************************************************
6810            ON TIMEOUT SC(@Gmacs),5 GOTO Err
6820            ON INTR SC(@Gmacs) GOTO Err
6830            ENABLE INTR SC(@Gmacs);2
6840            OUTPUT @Gmacs;"ST,"&VAL$(IVAL("01101100",2))
6850            OUTPUT @Gmacs;"MS,"&VAL$(Channel)&","&VAL$(Revolution)&","&VAL$(
Stop)
6860            OFF TIMEOUT
6870            OFF INTR
6880            SUBEXIT
6890 Err:       CALL Err_check(@Gmacs,"MS")
6900       SUBEND
6910       !
6920       !
```