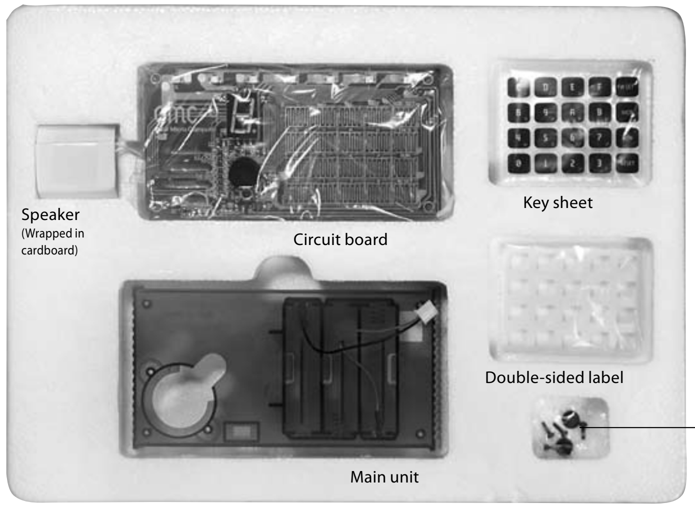
How to assemble the supp

GMC-4

4-bit Micro Computer

Assembly time: Approximately 15 minutes
(The electric circuit comes pre-assembled on the circuit board, so there is no need for soldering.)

Parts in the Kit



Things you will need

A Phillips screwdriver (#1), three new AA alkaline batteries or zinc-carbon batteries
(* NiCd and other rechargeable batteries have low voltage. Therefore the supplement may not operate when these batteries are used. batteries have high voltage. Using these batteries may break the circuit.)

Notes for tightening screws

The types of screws used for the supplement are those that carve grooves into the plastic as they are inserted (self-threading). The screwdriver most suited to tightening the screws is the #1 JIS screwdriver. When tightening the screws, firmly press the screwdriver straight against the screw and turn. It is said that 70 percent of the force applied is used for pushing against the screw and 30 percent for turning it. Precision screwdrivers are hard to turn, so use a small screwdriver with a grip diameter of about 2 cm.

Full-scale image of a #1 screwdriver

CAUTION

Please be sure to read the following instructions before assembling this kit.

- Take necessary caution when handling parts with pointed edges. There is a risk of injury.
- This kit includes screws and other small parts. Be careful not to swallow them. There is a risk of suffocation.

Three AA batteries are used. Incorrect use of the batteries may cause the generation of heat, explosions or liquid leakage. The following precautions should be taken.

- Do not use rechargeable batteries such as NiCd batteries, or Oxyride batteries
- Ensure that the positive and negative terminals of the batteries are aligned correctly.
- If liquid that leaked from the batteries gets into your eyes, rinse them well with plenty of water and consult a doctor immediately. If liquid leaks onto your skin or clothes, wash it off immediately.
- Always remove the batteries after use.
- Do not mix old and new batteries.

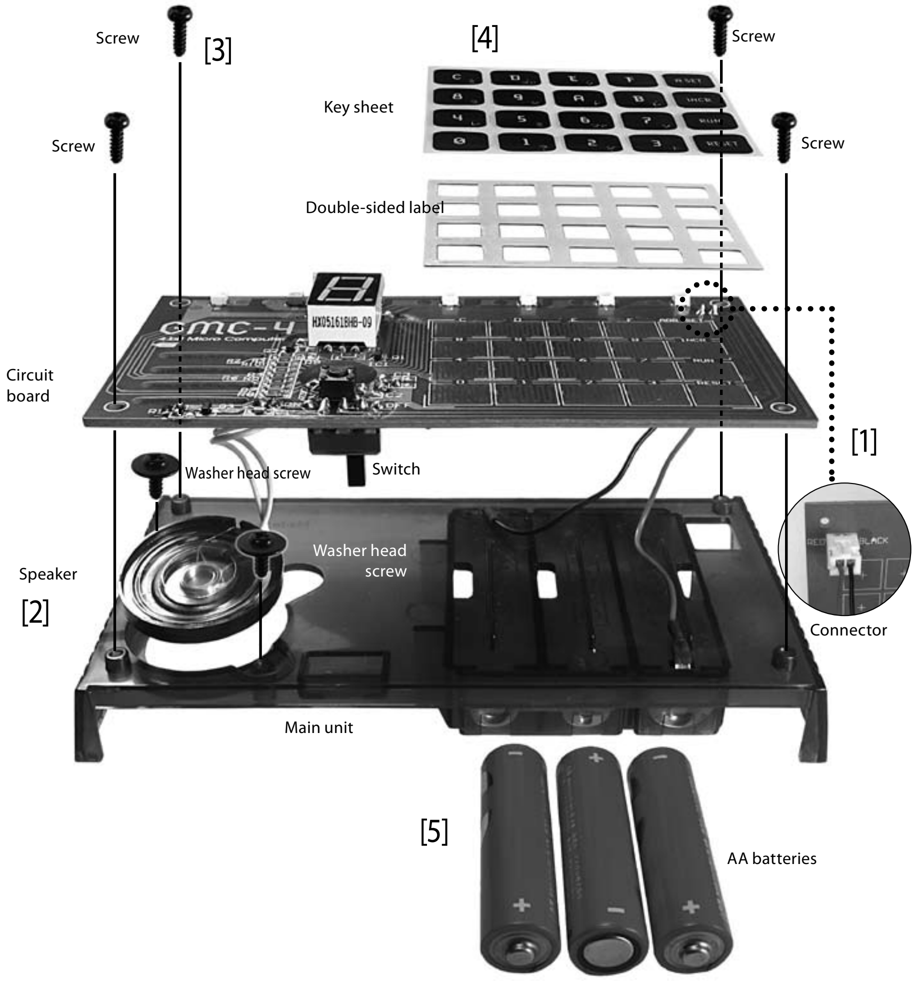
●Plastic materials used in this kit
Main unit (transparent black): ABS
Key sheet: PET
Circuit board: PH

●Metallic materials used in this kit
Screws: Iron

* Please dispose of this product in accordance with local regulations.

Assembling the Main Unit

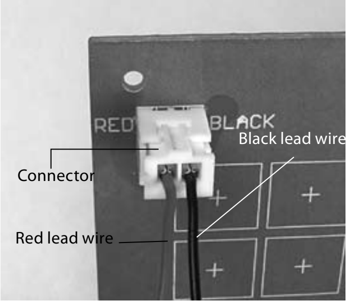
Attach the speaker, circuit board, etc. to the main unit in the order of steps [1] to [5].



[1] Connect the battery connector

Insert the connector for the main unit (battery box) into the plug on the circuit board.

* Be sure to insert the connection in the right direction.

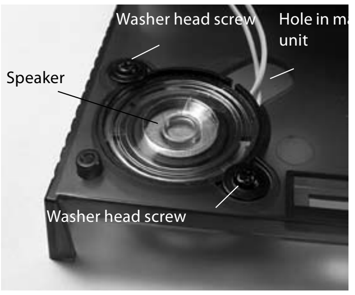


* Pass the speaker's lead wires through the whole on the main unit.

[2] Attach the speaker

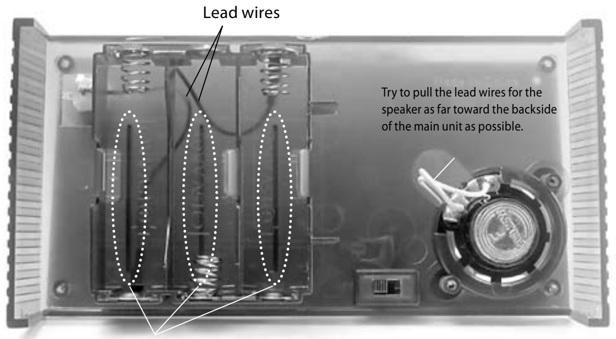
Attach the speaker to the hole on the main unit, and secure with washer head screws.

* Try not to touch the transparent cone when possible. If you accidentally touch the cone, causing it to collapse, lightly attach some cellophane tape to try to pull out and fix the indentation.



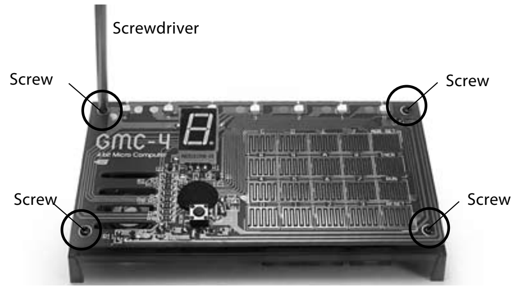
[3] Screw the circuit board onto the main unit

Being careful not to let the lead wires get caught between the circuit board and the main unit, affix the circuit board to the main unit with screws.



Try to pull the lead wires for the speaker as far toward the backside of the main unit as possible.

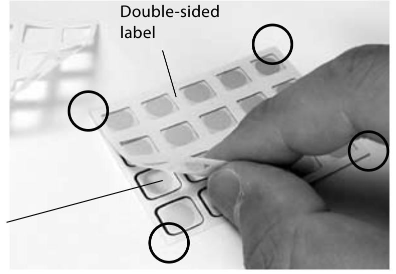
Pass the black and red lead wires around the protrusions in the three rows on the main unit.



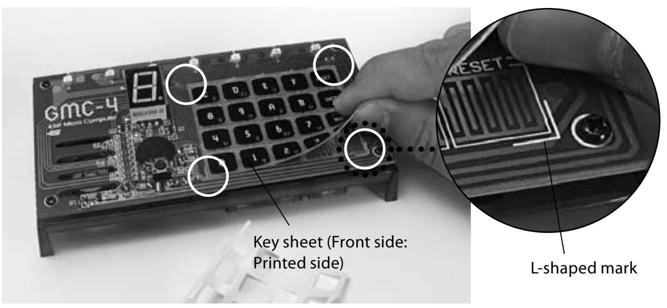
* Be careful not to damage the circuit with the screwdriver.

[4] Attach the key sheet to the circuit board

1. Remove the white paper on one side of the double-sided label, and place label over the back side of the key sheet so that the four corners line up.

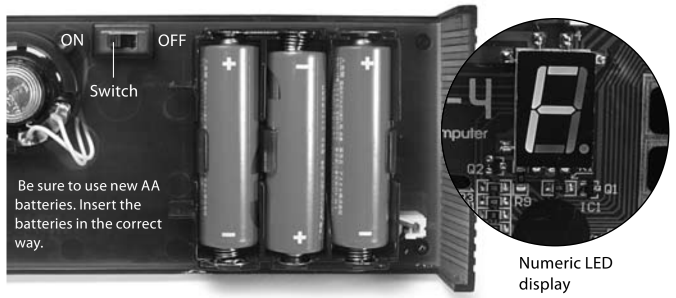


2. Remove the white paper on the other side of the double-sided label, and attach the key sheet to the circuit board by lining it up with the L-shaped



[5] Insert the batteries and turn on the switch

1. Insert batteries onto the battery box in the main unit and turn on the switch. Confirm that "F" is displayed on the numeric LED display.

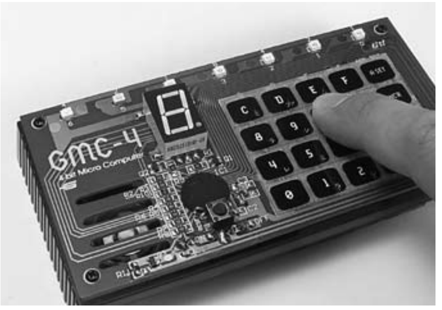


Be sure to use new AA batteries. Insert the batteries in the correct way.

2. Confirm that a short beep sounds when a key is pressed and the number or letter of the key pressed is displayed on the numeric LED display.

Instructions on how to use the supplement start from the following page.

* Be careful not to touch any of the electronic parts on the circuit board directly with your hand. Doing so may cause operation to become unstable. Also, the solder may protrude out. Therefore there is a risk of injury.



Q&A

Q: Nothing is displayed on the numeric LED display. No sound is produced.

A: Check the assembly condition.
Ensure that the connector for the battery box is fully engaged with the plug on the circuit board.
Ensure that the connector has been inserted in the right direction.

A: Ensure that you are using new batteries.
Replace any batteries that have run out of power. Be sure to check the orientation of the batteries.
Whenever the unit is not in use, be sure to turn it off and remove the batteries.

A: Check the speaker wires.
If either of the wires has become detached from the speaker, rectify the situation by resoldering.

A: Check the switch.
Ensure that the switch is turned to ON.

A: Check that the key sheet is not out of position.
If it is, carefully realign it.

Gakken Micro Computer

Supplement Micro Computer GMC-4 Manual

This supplement kit recreates the operation of the "Denshi-Block FX Micro Computer" that originally went on sale in 1981. The same command codes that were used for the original Micro Computer are used for all commands except for those for controlling external output.

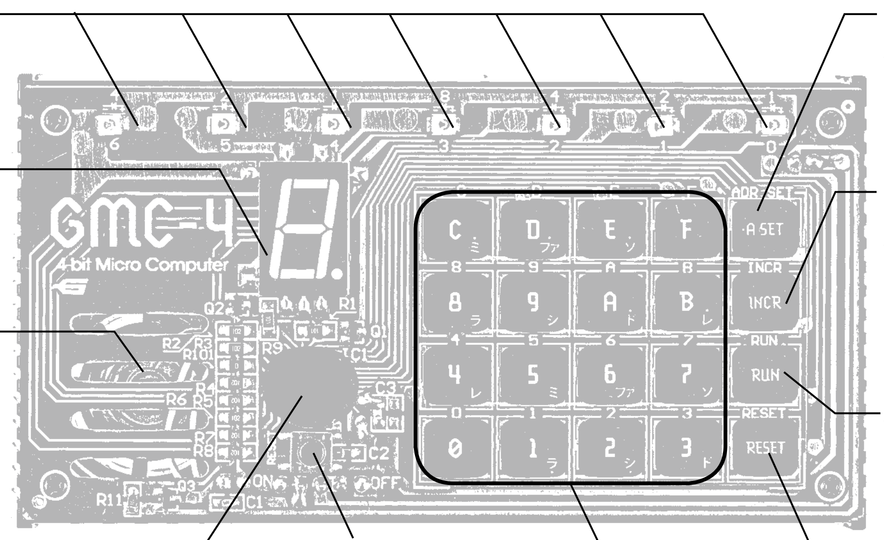
☒ CAUTION

● The 4-bit Micro Computer is an extremely delicate and complex electronic device. GAKKEN cannot guarantee that all operations work perfectly. Please be aware that GAKKEN will not be able to assist you except for in cases in which the Micro Computer fails to operate at all. In addition, although GAKKEN checked the operations of all registered programs, there may still be some possibility of malfunction as is the case with regular software.

● If operations for writing, etc. become unstable while you are experimenting with the supplement, replace the batteries with new ones, even if the LEDs are lit up.

Manual and programs created by Takuya Matsubara

1 Part Names



Binary LEDs
7 LEDs lined up along the side.
These LEDs are used to express a 7-digit binary number.

Numeric LED display
This is a single-digit, 7-segment LED display. It is used to display letters and numbers.

Speaker
Outputs sounds to check input, tones, etc.

CPU
Central processing unit. For details, please refer to page 48.

Hardware reset switch
The key for returning the status to what it was when the power was turned on.
Pressing this switch formats the contents of the memory.

Numeric keys
The keys for writing data.
There are a total of 16 keys.

A SET (address set) key
The key for setting the address.

INCR (increment) key
The key for writing data.

RUN (run) key
The key for executing programs.

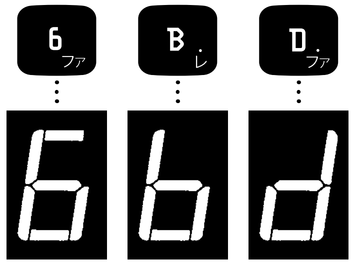
RESET (reset) key
The key for stopping the program and returning it to the address.

2 Operation Test

Test the GMC-4 to see if it is operating properly.
First, turn the "main switch" on the backside of the main unit to ON.
After the power is turned on, the numeric LED display turns on.

Key input test

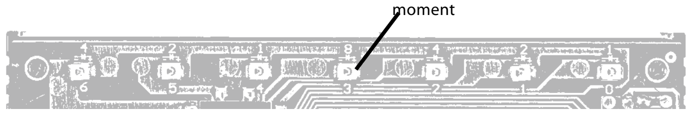
The 4 ×4 number of keys on which 0 to 9 and A to F are written are "Numeric keys." Try pressing each numeric key, one at a time, in order. When you press a numeric key, the letter or number for that key will be displayed on the "Numeric LED Display." For the alphabet keys, lowercase letters are displayed for B (b) and D (d) only.



The number "6" and the letters "b" and "d" differ as shown at right.

《Speaker test》

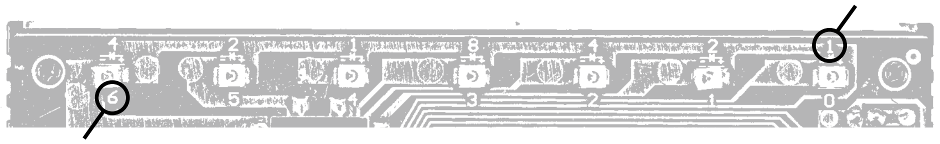
A short beep sound will be emitted from the speaker in conjunction with key input. At the same time, the center binary LED (fourth LED from the right) will also flash for a brief moment.



This LED only will flash for a moment

2 Operation Test

《Binary LED test》



The numbers written below each LED are the bit numbers.

The numbers written below each LED are the bit numbers.

The binary LEDs are used to display "binary numbers." Numbers are given for binary LEDs 0 through 6, and these numbers are referred to as bit numbers. If an LED is lit up, then that LED is displaying a "1," and if an LED is off, then that LED is displaying a "0."
Repeatedly pressing the INCR key changes the binary LED display in the order given below.

Binary LED display and correspondence with hexadecimal and decimal numbers

Binary LED display (●=OFF/○=ON)	Binary numbers	Hexadecimal numbers	Decimal numbers
●●●●●●●	0000000	00	0
●●●●●●○	0000001	01	1
●●●●●○●	0000010	02	2
●●●●●○○	0000011	03	3
●●●●○●●	0000100	04	4
●●●●○●○	0000101	05	5
●●●●○○●	0000110	06	6
●●●●○○○	0000111	07	7
●●●○●●●	0001000	08	8
●●●○●●○	0001001	09	9
●●●○●○●	0001010	0A	10
●●●○●○○	0001011	0B	11
●●●○○●●	0001100	0C	12
●●●○○●○	0001101	0D	13
●●●○○○●	0001110	0E	14
●●●○○○○	0001111	0F	15
●●○●●●●	0010000	10	16
⋮	⋮	⋮	⋮
○○○○○○○	1111111	7F	127

Key word

What are binary and hexadecimal numbers?

Binary numbers are numbers where the number of digits is increased by one to go from "1" to the next number "10." The numbers that we usually use in our daily lives are "decimal numbers." Decimal numbers are numbers where the number of digits is increased by one to go from "9" to the next number "10."
Meanwhile, hexadecimal numbers are numbers in which the number of digits is increased by one to go from "15" to the next number.
Numbers corresponding to 10 through 15 in the decimal number system are expressed by the letters A through F. Hexadecimal and binary numbers are compatible with each other, so they are used extensively in the computer industry.

How to read binary and hexadecimal numbers

Binary numbers are read in order from the left as a series of zeroes and/or ones. For hexadecimal numbers, the numbers written above each binary LED are used, and digits are divided into groups of three and four to be read. In the below example, the numbers are not read as "twelve" but rather as "one/two."

4	2	1	8	4	2	1	
●	●	○	●	●	○	●	(Binary LED)
Zero	Zero	One	Zero	Zero	One	Zero	(Reading of binary number)
		1			2		(Hexadecimal number)
		One			Two		(Reading of hexadecimal number)

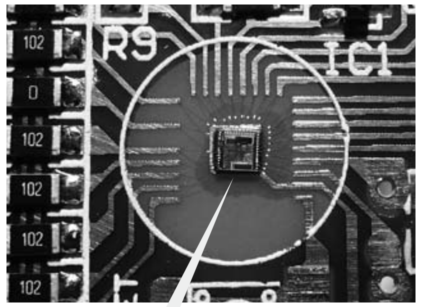
How to convert binary numbers to decimal numbers

The central processing unit (CPU), the heart of the Micro Computer, is divided into a number of different regions, depending on the role played. Terms such as address, register, etc. will be used throughout this supplement manual, so try to memorize their meanings here.

○	●	○	○	○	●	●	(Binary LED)
6	5	4	3	2	1	0	(Bit number=n)
2^6	2^4	2^3	2^2			(2n)	
64		+16	+8	+4			= 92(Decimal number)

3 How the GMC-4 Works

The central processing unit (CPU), the heart of the Micro Computer, is divided into a number of different regions, depending on the role played. Terms such as address, register, etc. will be used throughout this supplement manual, so try to memorize their meanings here.



Central processing unit (CPU)

The CPU is made by layering of silicon at a precision of several microns to several tens of microns and is also referred to as a "chip."

This is a picture of the chip before it is sealed with resin. You can see that the chip is wired directly into the circuit board with solder.

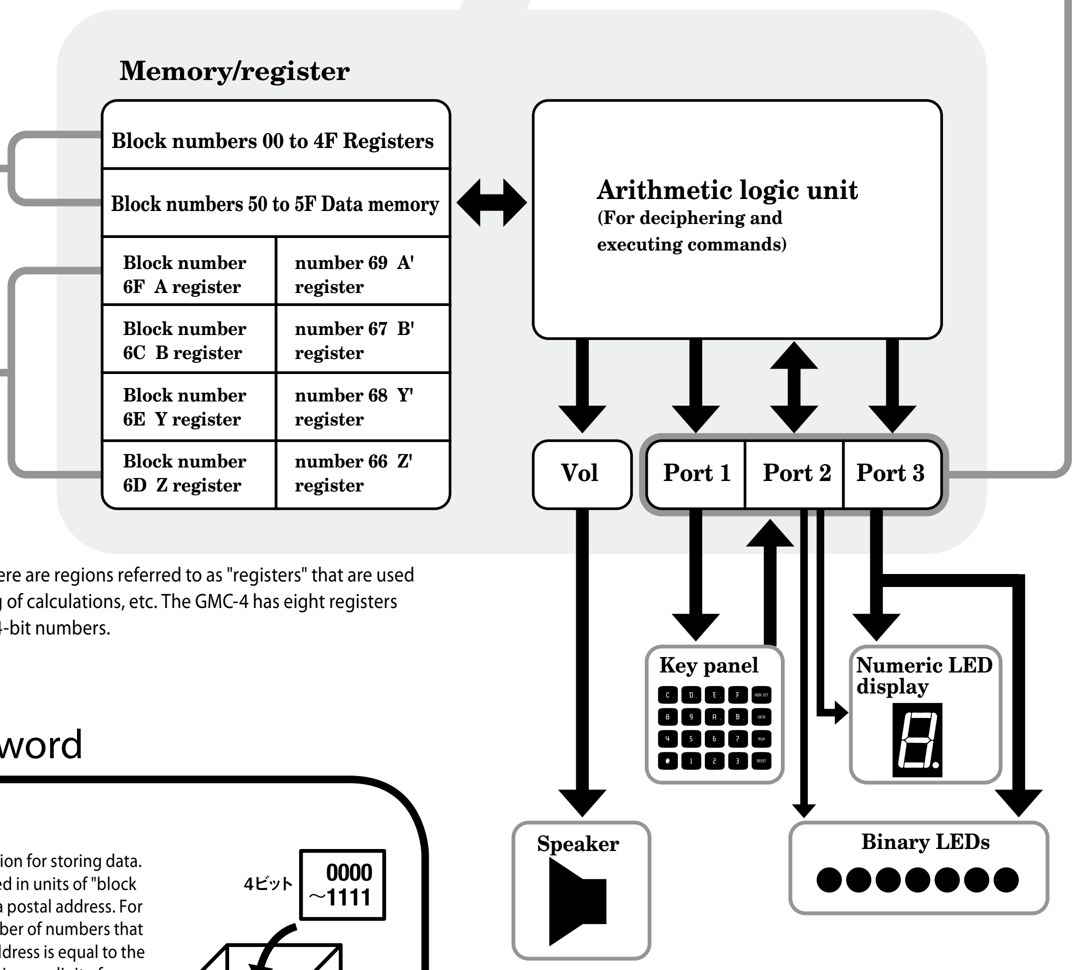
Memory

Regions and functions used to store data are referred to as "memory." Data stored in the memory of the GMC-4 is lost when the power is turned off. The Micro Computer in this supplement has the following types of memory.

Address	Memory functions
00 to 4F	Program memory (memory used to store programs)
50 to 5F	Data memory (memory used to store calculation results, etc.)
66 to 6F	Register (memory used to temporarily store data midway through a calculation, with a total of 8 types)

Port

The part where signals are output and input is called a "port." Ports have inputs and outputs and are used for either of the two. In this supplement, ports are connected to the keys, LEDs, etc.–––


Memory/register
Block numbers 00 to 4F Registers
Block numbers 50 to 5F Data memory
Block number 6F A register
number 69 A' register
Block number 6C B register
number 67 B' register
Block number 6E Y register
number 68 Y' register
Block number 6D Z register
number 66 Z' register
Arithmetic logic unit
(For deciphering and executing commands)
Vol
Port 1
Port 2
Port 3
Key panel
Numeric LED display
Speaker
Binary LEDs

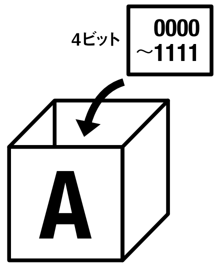
Register

In the Micro Computer, there are regions referred to as "registers" that are used temporarily for processing of calculations, etc. The GMC-4 has eight registers that can be used to store 4-bit numbers.

Key word

Address

An address is a location for storing data. An address is handled in units of "block number x," just like a postal address. For the GMC-4, the number of numbers that can be stored per address is equal to the number of numbers in one digit of a hexadecimal number. When you want to save an even larger number, the number will be stored across multiple addresses.



◆Dynamic lighting

Dynamic lighting is a lighting method in which the lights are switched on and off at high speed. Dynamic lighting is used in this 4-bit Micro Computer to turn the binary LEDs and numeric LED display on in an alternating manner. The cycle for lighting is about 4 ms (1/1000 of 4 s), and it appears that both the binary LEDs and the numeric LED display are on due to the afterimage effect on the eye. Through use of dynamic lighting, the number of output ports needed was reduced, and the circuit could be simplified.

4 How to Write, Check, and Revise Data

Try writing some data to the GMC-4. The data is written to an address.
An address is handled in units of "block number X," just like a postal address.

Writing data ≫

Data can be written to addresses from block number 00 to block number 5F. Hexadecimal numbers are inserted into the addresses in order, starting from block number 00. Here, as an example, try writing "1A" to block number 00 to block number 02. "Commands" are assigned to each piece of this data, and those commands are executed on each piece of the data in order of address. For a list of commands, please see page 56.

Address	Data
00	1
01	A

Writing "1A" to addresses block number 00 to block number 02

First, press the hardware reset key to clear all data.
Once cleared, all data will be F.

(●=OFF/○=ON)

Key to press	Binary LED (address)	Numeric LED (content)	
Press the RESET key	●●●●●●●	F	The binary LEDs display the address, and the numeric LED display displays the content. Pressing RESET will make the address 00. In this case, it means that "data of F has been inserted into block number 00."
↓			
Press the 1 key	●●●●●●●	1	The number 1 is displayed on the numeric LED display, and the data is overwritten. However, the written data is still in a conditional approval state. Next, press the INCR key to designate the data.
↓			
Press the INCR key	●●●●●●○	F	The number 1 has been written to memory block number 00. One will be added to the address automatically, and content of F, which had been inserted into block number 01, will be displayed on the numeric LED display.
↓			
Press the A key	●●●●●●○	A	The letter A is displayed on the numeric LED display. Press the INCR key to designate the data.
↓			
Press the INCR key	●●●●●○●	F	Data input is now completed. Content of F, which was inserted into block number 02, is displayed on the numeric LED display.

4 How to Write, Check, and Revise Data

□□ Checking data ››

Without inputting data, press INCR only to add one to the address sequentially. Try pressing keys in the order of RESET , INCR and INCR ,. Data that you just entered should be displayed on the numeric LED display in order of 3, 8, D, and C. Check the content of the data as described. The reason why RESET is to be pressed at the beginning is to return the address to block number 00.

□□ Revising data ››

By pressing the A SET key after entering the address block number, you can jump to the address of the memory to be read. For example, let's say that there is a mistake in the data in address block number 50. Press the keys in the order given below.

Press RESET
The content of the memory for block numer 00 is displayed on the numeric LED display.
↓
Press the 5 key
The number 5 is displayed on the numeric LED display. The written data is still in a conditional approval state.
↓
Press the 0 key
The number 0 is displayed on the numeric LED display. At this point, the two-digit number, "50," has actually been input.
↓
Press the A SET key
The content of the memory for block numer 50 is displayed on the numeric LED display.
The number "50," which is the address, is displayed by the binary LEDs.
Entering the new data and then pressing INCR to designate the data completes the revision of the data.

□□ Clearing data ››

Pressing the hardware reset switch turns all data into "F" to clear the memory. You can also clear the memory by turning the main switch on the main unit to OFF. Pressing the RESET key only allows you to jump to the address for block number 00 without clearing the content of the memory.

5 Program execution types

Execution of programs can be roughly divided into two modes of RUN and STEP. Along with the display of binary numbers and the presence or absence of a key input sound, there are four types in total.

Key input	Mode	Address display on binary LEDs	Key input sound
RESET 1 RUN	RUN mode	No	No
RESET 2 RUN	RUN mode	Yes	No
RESET 5 RUN	STEP mode	No	No
RESET 6 RUN	STEP mode	Yes	Yes

The "RUN mode" is a mode for executing programs normally. Programs that have been written can all be executed at once, from block number 00 to the end.
The "STEP mode" is a mode for executing commands one by one and is used mainly for checking program operations. Commands proceed one by one each time that the INCR key is pressed.

6 Trying Out Sample Programs

The GMC-4 comes with seven different types of sample programs written to it. You can immediately start playing games, etc. with the GMC-4, even without writing any programs yourself.

Sample program 1: Electronic organ
Sample program 2: Guess the Music Notes game
Sample program 3: Hit the Mole game
Sample program 4: Tennis game
Sample program 5: Timer
Sample program 6: Automatic musical performance
Sample program 7: Automatic transmission of telegrams in Morse code

(Morse code for telegrams is introduced on the GAKKEN website: http://otonanokagaku.net/)

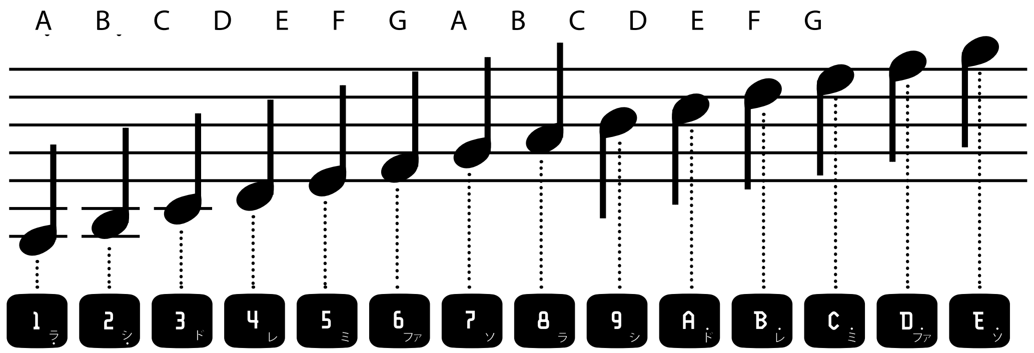
Sample program 1 Electronic organ

Try making some electronic sounds.

(1) Starting the program
Turn the main switch on, and press the RESET , 9 and RUN keys in the order given.

(2) Performing with music
Pressing the 1 to E keys will produce a sound.

Each key is assigned to a different note on the musical scale.
The sound will be made only while the key is being pressed.



Sample program 2 Guess the Music Notaes game

This is a game where you have to guess what note the GMC-4 is outputting. Try testing the limits of your sense of pitch and memory skills.

1) Starting the game

Turn the main switch on, and press the RESET , B , and RUN keys in the order given. Inputting keys starts the game.

2) How to play the game

A melody will play from the speaker. Once the melody has stopped, try to push the 1 to E keys in the same order of notes. The melody always starts from "C." So you should always press the 3 (C) key first. If you press the keys in the right order, the number of notes in the melody will increase by one at a time, and the game will continue. If you make a mistake and press the wrong key, the game will finish, and your final score will be displayed on the numeric LED display.

If you get all ten notes right, it will be displayed as such on the numeric LED display .

After you finish the game, press the RUN key to play again.

Sample program 3 Hit the Mole game

This is a Hit the Mole game that uses the binary LED display. Try to beat the game with how sharp your reflexes are.

1) Starting the game

After turning the main switch on, press the RESET , C , and RUN keys. Next, choose the speed at which the moles will appear using the following keys. This determines the difficulty level for the game.

2) How to play the game

One of the binary LEDs will light up, so press one of the 0 to 6 keys to match the lit LED. For example, in the following case……

● ● ○ ● ● ● ●

【6】【5】【4】【3】【2】【1】【0】

(Look at the numbers written above the LEDs) (●=OFF/○=ON)

Pressing the 4 key is the correct answer.
If you press the right key, a short beep sound will go off, and your score will increase by one. If you press the wrong key, or if you fail to press the right key within the allocated amount of time, your answer will be counted as wrong.

3) Ending and restarting the game

Once a total of ten moles have appeared, the game will end. Your final score will be displayed on the numeric LED display. If you get a score of 10, it will be displayed as such.

Pressing the RUN key will allow you to restart the game.

Sample program 4 Tennis game

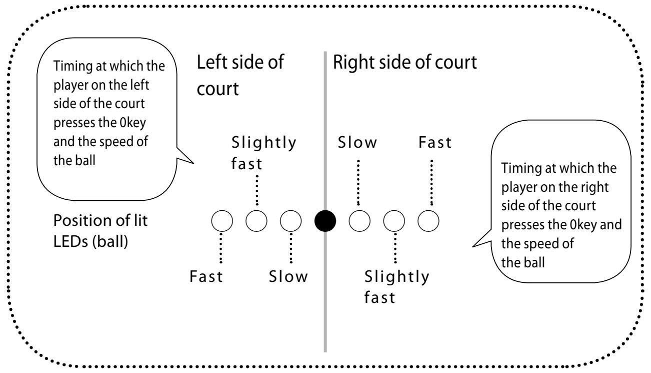
his is a tennis game that can be played by two people.

1) Starting the game

Turn the main switch on, and press the RESET , D , and RUN keys in the order given. Pressing the keys starts the game.

2) How to play the game

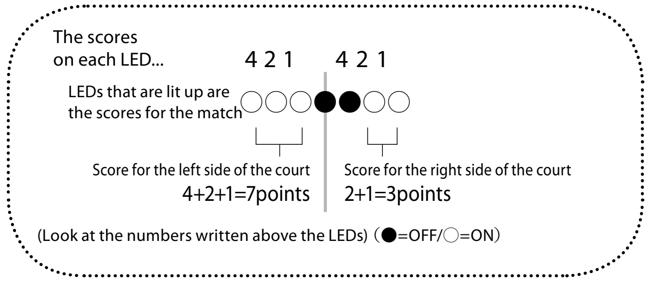
The binary LEDs are set up to look like a tennis court.


Timing at which the player on the left side of the court presses the 0key and the speed of the ball
Left side of court
Right side of court
Slightly fast
Slow
Fast
Position of lit LEDs (ball)
Fast
Slow
Slightly fast
Timing at which the player on the right side of the court presses the 0key and the speed of the ball

Press the 0 and 3 keys to hit the ball. The 0 key is for the player on the left side of the court, and the 3 key is for the player on the right side of the court.
The speed of the ball being hit back will change depending on the timing at which the keys are pressed.

3) Ending the game

If the score of either player reaches seven points, the game will end, and the final scores will be displayed on the binary LEDs. For example, in the following case……


The scores on each LED...
4 2 1
4 2 1
LEDs that are lit up are the scores for the match
Score for the left side of the court
4+2+1=7points
Score for the right side of the court
2+1=3points
(Look at the numbers written above the LEDs) (●=OFF/○=ON)

……This yields a "win, 7 to 3, awarded to the player on the left side of court."
Pressing the RUN key will allow you to restart the game.

Sample program 5 Timer

This timer can be used to set a time limit up to 7 minutes and 59 seconds, maximum. When the time remaining on the timer goes down to 0, a sound will be emitted from the speaker.

1) Setting the timer

Write the time for the timer in block number 00 to block number 02 in the program memory.

Address	of data
00	Minutes (0 to 7)
01	Seconds, tens digit (0 to 5)
02	, ones digit (0 to 9)

Key to press
RESET
3 INCR
2 INCR
5 INCR

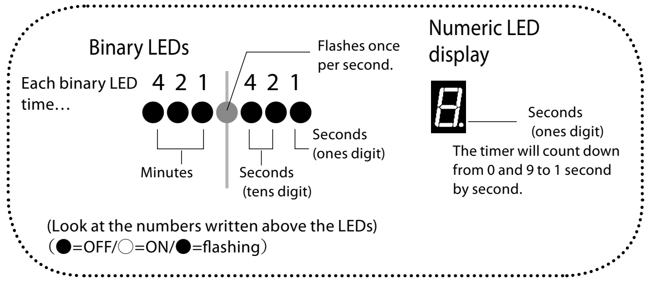
For example, if you want to set the timer to 3 minutes and 25 seconds, input the keys in the order given on the left.

Lastly, press RESET , E , and RUN in the order given to start the timer.

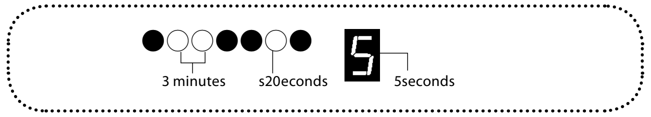
2) Counting down on the timer

A short beep sound will be emitted from the speaker every second while the timer is in operation.

The time remaining on the timer will be displayed on the binary LEDs and numeric LED display.



For example, a display like the one shown below indicates "3 minutes and 25 seconds."



3) Ending the timer

When the time remaining on the timer goes to zero, a sound will be emitted, and the timer will stop.

Sample program 6 Automatic musical performance

This program is for performing music automatically. You can input performance data to let it play your favorite melodies.

1) Inputting performance data

Turn the main switch on, and write performance data to the program memory. You can store performance data in blocks from block number 00 to block number 5F.

Set the tempo of the song to the first memory block, block number 00.

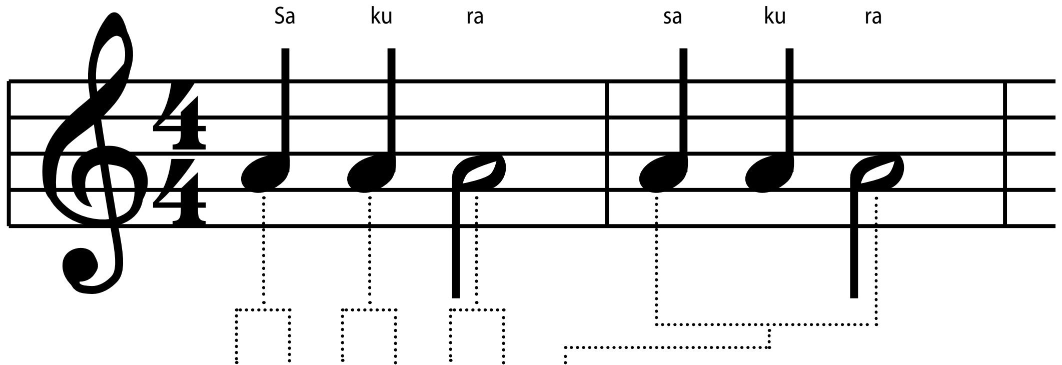
You can input up to 46 musical notes into the performance data.

The data for lengths of notes and rests is as shown on the right.

Musical note code

Length of notes and rests	Data
Sixteenth note, sixteenth rest	0
Eighth note, eighth rest	1
Dotted eighth note	2
Quarter note, quarter rest	3
Dotted quarter note	5
Half note, half rest	7
Dotted half note	B
Whole note, whole rest	F

For example, inputting part of the song "Sakura Sakura" into the performance data yields the following.



the value increases, the slower the tempo will get.

Note data		8(A)		8(A)		9(B)		
Data for length of note			3		3		7	0 Repeats
Tempo of song (0〜F)	A							
Address	00	01	02	03	04	05	06	07

To input data for "Sakura Sakura" into the memory, press each of the keys one by one in the order given below.

Key to press	Binary LEDs (●=OFF/○=ON)	Numeric LED display
RESET	●●●●●●●	?
A	●●●●●●●	A
INCR	●●●●●●●	?
8	●●●●●●●	8
INCR	●●●●●●●	?
3	●●●●●●●	3
INCR	●●●●●●●	?
8	●●●●●●●	8
INCR	●●●●●●●	?
3	●●●●●●●	3
INCR	●●●●●●●	?
9	●●●●●●●	9
INCR	●●●●●●●	?
7	●●●●●●●	7
INCR	●●●●●●●	?
F	●●●●●●●	F
INCR	●●●●●●●	?
0	●●●●●●●	0
INCR	●●●●●●●	?

The "?" means that any number would work for that output.
The data that had been entered into the address will be displayed.

2) Checking the data

Check the written data. You can check the data that you entered by pressing RESET , INCR , INCR , INCR ………

3) Starting the performance

Press RESET , A , and RUN to start the performance.

4) Revising the data

If the wrong data has been input, you can revise only the address with the wrong data, without having to go back to the beginning to write new data.

For example, an 8 has been written to address 03. (The correct data should be a 3)

RESET

0
3 Address block number for wrong data

8 Press the button to overwrite. This completes the revision of the data.

5) Stopping and restarting the performance

Pressing the RESET key stops the performance. Press A and RUN when you want to restart the performance.

6) Changing the tempo

The tempo for the song is set in the very first memory block, block number 00. As the value increases from 0 to F, the tempo will get slower.

Press the RESET , 0 , and INCR keys for the fastest performance.

7) Playing a song you like

First, change the notes and rests for the music you want to play into performance data and then input data as described below.

Input the tempo data to block number 00. (Input a number from 0 to F . The tempo will get slower as the numbers get bigger.)
Write either note data or rest data to the odd-numbered block numbers.
Write data for the length of notes or rests to the even-numbered block numbers.
Write an F if you wish to end the melody and a 0 if you wish to repeat the performance after the end of the performance data.

7 Command Codes and Program Structure

In order to operate the computer, you will need to use "Commands." You can make programs by combining those commands. Incidentally, commands are assigned by the CPU in advance.

Writing data ››

The GMC-4 comes with a total of 30 different types of commands. The types of commands include ones for "calculating," "turning LEDs on," "jumping," etc. These commands are combined in complex ways to create programs. For details on the commands, please check the "List of Command Codes" on the following page.
"Command symbols" are symbols used to express commands "in language that people can understand." "Command codes" are translations of these command symbols "into language that a CPU can understand."
Hexadecimal forms are used for the command codes; for example, translating the command symbol "TIA 1" would yield a command code of "81" in block number 2 of the memory. The amount of memory that a command code requires depends on the type of command.

Program structure ››

You can create a program merely by combining the three types of structures of "sequential processing," "branch processing," and "repetitive processing."
This structure can be represented as the flowchart shown below. A "flowchart" is a diagram that shows the flow of processes in a system.

PROGRAM

Program that allows you to experience what commands are

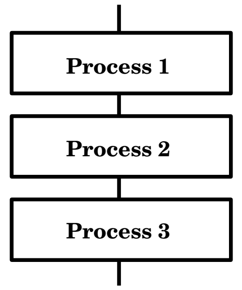
Numeric LED display lighting experiment uthe TIA and AO commands

This is one of the most basic examples of a program. In order to merely display the number 5 on the numeric LED display, we will use two commands: "TIA" and "AO."

Address	Command symbols	Command code	Command operations
00	TIA	**8**	Inputs the number 5 into register A
01	●5●	**5**	
02	AO	**1**	Outputs the content of register A to a port and turns the numeric LED display on.

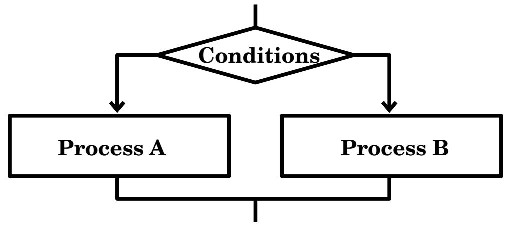
After you've written the program, execute it by pressing RESET, , and . The number is displayed on the numeric LED display.
If you change the number input to address 01, the number displayed on the numeric LED display will change.

Sequential processing



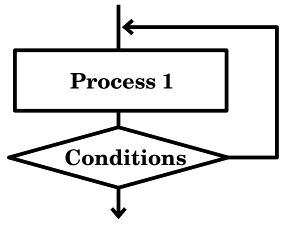
Process 1, Process 2, and Process 3 are executed in sequential order.

Branch processing

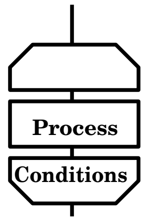


The diamond shape indicates the branch instruction. Process is divided into branches in accordance with certain conditions. However, no matter how much a process is divided up, the flow must always return to one place. By setting up one start and one end for each process, you can make it difficult for malfunctions to occur, even for complex programs. Incidentally, branches are managed using markers called execution flags that demonstrate the results of the command.

Repetitive processing



Depending on the conditions, the flow returns to a previous process.
The flow proceeds to the following process once certain conditions have been fulfilled.



You can also write the flowchart like this.

Key word

Execution flags

If you want to program a branch process, use the "JUMP" command. The JUMP command changes the block number to be executed. The JUMP command has the characteristic of "making the program jump when the execution flag is 1." An "execution flag" is a marker that indicates the results of the execution of a command. Conditions for changing the execution flag depend on the command (refer to the "List of Command Codes"), but by combining the CIA command and the JUMP command, for example, you can execute a branch process to "make the program jump when register A is not a certain value." There are also commands that will set the execution flag to 1 without fail after execution. If you combine that command and the JUMP command together, you can create a process that forces the program to jump.

List of GMC-4 Command Codes

Command code	Command symbol	Operation	Execution flag	Description
0	KA	K→Ar	0,1	Substitutes the number of the pressed numeric key (0 to F) into register A. The execution flag is 1 when no key is pressed and 0 if a key is pressed.
1	AO	Ar→Op	1	Lights up LEDs for the value in register A (0 to F) on the numeric LED display.
2	CH	Ar⇔Br, Yr⇔Zr	1	Switches the values in register A and register B around with each other and also switches the values in register Y and register Z around with each other.
3	CY	Ar⇔Yr	1	Switches the values in register A and register Y around with each other.
4	AM	Ar→M	1	Substitutes the value in register A into the data memory.
5	MA	M→Ar	1	Substitutes the value in the data memory into register A.
6	M+	M+Ar→Ar	0,1	Adds a value to the value in the data memory of register A. If a number is carried over to the next digit, then the execution flag is 1. If not, the execution flag is 0.
7	M-	M-Ar→Ar	0,1	Subtracts the value in register A from the value in the data memory and substitutes the results into register A. If the value is negative, then the execution flag is returned as 1. If not, then the execution flag is 0.
8	TIA □	□→Ar	1	Substitutes a designated value (0 to F) into register A.
9	AIA □	Ar+□→Ar	0,1	Adds a designated value (0 to F) to register A. If a number is carried over to the next digit, then the execution flag is 1. If not, then the execution flag is 0.
A	TIY □	□→Yr	1	Substitutes a designated value (0 to F) into register Y.
B	AIY □	Yr+□→Yr	0,1	Adds a designated value (0 to F) to register Y. If a number is to be carried over to the next digit, the execution flag is 1. If not, the execution flag is 0.
C	CIA □	Ar≠□?	0,1	If the value in register A is the same as ≤, then the execution flag is 0. If the values are not the same, the execution flag is 1.
D	CIY □	Yr≠□?	0,1	If register Y is equal to ≤, then the execution flag is 0. If the values are not the same, then the execution flag is 1.
F	JUMP □□	□□→PC	1	If the execution flag is 1, then the program jumps to a specified address. If the execution flag is 0, then the program continues on as written.

E0	CAL RSTO	1		Turns LEDs on the numeric LED display off.
E1	CAL SETR	1		Turns on one of the binary LEDs. Register Y = bit number (0 to 6) of the binary LED.
E2	CAL RSTR	1		Turns off one of the binary LEDs. Register Y = bit number (0 to 6) of the binary LED.
E4	CAL CMPL	1		Inverts the value of 0 or 1 in register A (bit inversion).
E5	CAL CHNG	1		Switches the values in registers A/B/Y/Z and registers A'/B'/Y'/Z' (auxiliary registers) around with each other.
E6	CAL SIFT	0,1		Shifts the value in register A one bit to the right. If the original value is an even number, then the execution flag is 1. If it is an odd number, then the execution flag is 0.
E7	CAL ENDS	1		Produces the end sound.
E8	CAL ERRS	1		Produces an error sound.
E9	CAL SHTS	1		Produces a short beep sound.
EA	CAL LONS	1		Produces a long beep sound.
EB	CAL SUND	1		Produces the sound of a note (1 to E) designated by register A.
EC	CAL TIMR	1		Waits for processing only for a length of time designated by register A. The length of the wait time is {(Value in register A + 1) ×0.1 s}.
ED	CAL DSPR	1		Lights up the binary LED of the value in the data memory. Stores the data for the top three bits in block number 5F and for the bottom four bits in block number 5E.
EE	CAL DEM-	1		Subtracts the value in register A from the value in the data memory. Stores the results of the calculation in the memory after it has been converted to decimal number. After execution, 1 is subtracted from the value in register Y.
EF	CAL DEM+	1		Adds the value in register A from the value in the data memory. Stores the results of the calculation in the memory after it has been converted to decimal number. When a number is carried over to the next digit, the result is written to the data memory automatically. After execution, 1 is subtracted from the value in register Y.

Abbreviated symbol	Register name (address)
Ar	Register A (6F)
Br	Register B (6C)
Yr	Register Y (6E)
Zr	Register Z (6D)
M	Memory (50 to 5F)

Abbreviated symbol	Auxilliary register name (address)
A'r	Register A'r (69)
B'r	Register B'r (67)
Y'r	Register Y'r (68)
Z'r	Register Z'r (66)
●	A number from 0 to F is input

☒ CAUTION Commands such as CAL RSTO, CAL SUND, etc. are called "subroutines."
Subroutines are parts grouped together to make complex commands easier to use. There are 16 different types of subroutines, and the command symbols for all 16 types start with "CAL." Please be aware that subroutines will not execute if the execution flag is not 1.

8 Introduction to Programs

program 1 15-second counter

Program Binary LEDs（-=OFF/*=ON）

Binary LEDs	Address	Command symbols	Command code
-------	00	TIY	A
------*	01	1	1
-----*-	02	TIA	8
-----**	03	9	9
----*--	04	CAL	E
----*-*	05	TIMR	C
----**-	06	CY	3
----***	07	AO	1
---*---	08	CY	3
---*--*	09	CAL	E
---*-*-	0A	SHTS	9
---*-**	0B	AIY	B
---**--	0C	1	1
---**-*	0D	JUMP	F
---***-	0E	1	1
---****	0F	3	3
--*----	10	JUMP	F
--*---*	11	0	0
--*--*-	12	2	2
--*--**	13	CAL	E
--*-*--	14	ENDS	7
--*-*-*	15	JUMP	F
--*-**-	16	1	1
--*-***	17	5	5

program 2 Electronic dice

Binary LEDs	Address	Command symbols	Command code
-------	00	TIY	A
------*	01	1	1
-----*-	02	CY	3
-----**	03	AO	1
----*--	04	CY	3
----*-*	05	AIY	B
----**-	06	1	1
----***	07	CIY	D
---*---	08	7	7
---*--*	09	JUMP	F
---*-*-	0A	0	0
---*-**	0B	E	E
---**--	0C	TIY	A
---**-*	0D	1	1
---***-	0E	KA	0
---****	0F	JUMP	F
--*----	10	0	0
--*---*	11	2	2
--*--*-	12	JUMP	F
--*--**	13	0	0
--*-*--	14	E	E

program 3 Adding and subtracting single-digit hexadecimal numbers

Binary LEDs	Address	Command symbols	Command code
-------	00	TIY	A
------*	01	1	1
-----*-	02	MA	5
-----**	03	TIY	A
----*--	04	0	0
----*-*	05	M+	6
----**-	06	JUMP	F
----***	07	0	0
---*---	08	C	C
---*--*	09	JUMP	F
---*-*-	0A	1	1
---*-**	0B	0	0
---**--	0C	TIY	A
---**-*	0D	0	0
---***-	0E	CAL	E
---****	0F	SETR	1
--*----	10	AO	1
--*---*	11	JUMP	F
--*--*-	12	1	1
--*--**	13	1	1

CAL SETR is a command that makes the binary LEDs to light up.

program 4 Binary light show using bit shift operations

Binary LEDs	Address	Command symbols	Command code
-------	00	CAL	E
------*	01	DSPR	D
-----*-	02	TIA	8
-----**	03	0	0
----*--	04	CAL	E
----*-*	05	TIMR	C
----**-	06	TIY	A
----***	07	F	F
---*---	08	MA	5
---*--*	09	CAL	E
---*-*-	0A	SIFT	6
---*-**	0B	JUMP	F
---**--	0C	1	1
---**-*	0D	A	A
---***-	0E	AM	4
---****	0F	TIY	A
--*----	10	E	E
--*---*	11	MA	5
--*--*-	12	CAL	E
--*--**	13	SIFT	6
--*-*--	14	AIA	9
--*-*-*	15	8	8
--*-**-	16	AM	4
--*-***	17	JUMP	F
--**---	18	2	2
--**--*	19	1	1
--**-*-	1A	AM	4
--**-**	1B	TIY	A
--***--	1C	E	E
--***-*	1D	MA	5
--****-	1E	CAL	E
--*****	1F	SIFT	6
-*-----	20	AM	4
-*----*	21	KA	0
-*---*-	22	JUMP	F
-*---**	23	0	0
-*--*--	24	0	0
-*--*-*	25	TIY	A
-*--**-	26	F	F
-*--***	27	MA	5
-*-*---	28	AIA	9
-*-*--*	29	8	8
-*-*-*-	2A	AM	4
-*-*-**	2B	JUMP	F
-*-**--	2C	0	0
-*-**-*	2D	0	0

program 7 Random number music generator

Binary LEDs	Address	Command symbols	Command code
-------	00	AIY	B
------*	01	1	1
-----*-	02	M+	6
-----**	03	AM	4
----*--	04	CAL	E
----*-*	05	SUND	B
----**-	06	JUMP	F
----***	07	0	0
---*---	08	0	0

program 5 Analog lighting with the numeric LED display (PWM controls)

Binary LEDs	Address	Command symbols	Command code
-------	00	TIY	A
------*	01	1	1
-----*-	02	TIA	8
-----**	03	0	0
----*--	04	AM	4
----*-*	05	TIY	A
----**-	06	0	0
----***	07	CAL	E
---*---	08	RSTO	0
---*--*	09	MA	5
---*-*-	0A	AIA	9
---*-**	0B	2	2
---**--	0C	JUMP	F
---**-*	0D	1	1
---***-	0E	2	2
---****	0F	JUMP	F
--*----	10	0	0
--*---*	11	A	A
--*--*-	12	TIA	8
--*--**	13	0	0
--*-*--	14	AO	1
--*-*-*	15	MA	5
--*-**-	16	AIA	9
--*-***	17	E	E
--**--*	19	1	1
--**-*-	1A	6	6
--**-**	1B	TIY	A
--***--	1C	2	2
--***-*	1D	TIA	8
--****-	1E	1	1
--*****	1F	M+	6
-*-----	20	JUMP	F
-*----*	21	2	2
-*---*-	22	7	7
-*---**	23	AM	4
-*--*--	24	JUMP	F
-*--*-*	25	0	0
-*--**-	26	5	5
-*--***	27	AM	4
-*-*---	28	TIY	A
-*-*--*	29	1	1
-*-*-*-	2A	MA	5
-*-*-**	2B	TIY	A
-*-**--	2C	0	0
-*-**-*	2D	CIA	C
-*-***-	2E	0	0
-*-****	2F	JUMP	F
-**----	30	3	3
-**---*	31	C	C
-**--*-	32	TIA	8
-**--**	33	2	2
-**-*--	34	M+	6
-**-*-*	35	JUMP	F
-**-**-	36	4	4
-**-***	37	6	6
-***---	38	AM	4
-***--*	39	JUMP	F
-***-*-	3A	0	0
-***-**	3B	5	5
-****--	3C	TIA	8
-****-*	3D	2	2
-*****-	3E	M-	7
-******	3F	JUMP	F
*------	40	4	4
-----	41	6	6
-----	42	AM	4
*----**	43	JUMP	F
-----	44	0	0
----*	45	5	5
*---**-	46	TIY	A
*---***	47	1	1
-----	48	MA	5
----*	49	CAL	E
---*-	4A	CMPL	4
---**	4B	AM	4
*--**--	4C	JUMP	F
*--**-*	4D	0	0
*--***-	4E	5	5

program 6 Calculating a maximum number

Binary LEDs	Address	Command symbols	Command code
-------	00	TIY	A
------*	01	F	F
-----*-	02	TIA	8
-----**	03	2	2
----*--	04	AM	4
----*-*	05	CAL	E
----**-	06	CHNG	5
----***	07	TIY	A
---*---	08	0	0
---*--*	09	MA	5
---*-*-	0A	CH	2
---*-**	0B	TIY	A
---**--	0C	0	0
---**-*	0D	MA	5
---***-	0E	CAL	E
---****	0F	CMPL	4
--*----	10	AIY	B
--*---*	11	1	1
--*--*-	12	M+	6
--*--**	13	JUMP	F
--*-*--	14	2	2
--*-*-*	15	8	8
--*-**-	16	MA	5
--*-***	17	CH	2
--**---	18	AIY	B
--**--*	19	1	1
--**-*-	1A	CH	2
--**-**	1B	CH	2
--***--	1C	CAL	E
--***-*	1D	CHNG	5
--****-	1E	AIA	9
--*****	1F	1	1
-*-----	20	JUMP	F
-*----*	21	3	3
-*---*-	22	6	6
-*---**	23	CAL	E
-*--*--	24	CHNG	5
-*--*-*	25	JUMP	F
-*--**-	26	0	0
-*--***	27	D	D
-*-*---	28	MA	5
-*-*--*	29	AIY	B
-*-*-*-	2A	F	F
-*-*-**	2B	AM	4
-*-**--	2C	AIY	B
-*-**-*	2D	1	1
-*-***-	2E	CH	2
-*-****	2F	AIY	B
-**----	30	1	1
-**---*	31	AM	4
-**--*-	32	CH	2
-**--**	33	JUMP	F
-**-*--	34	1	1
-**-*-*	35	C	C
-**-**-	36	TIY	A
-**-***	37	F	F
-***---	38	MA	5
-***--*	39	AIA	9
-***-*-	3A	1	1
-***-**	3B	JUMP	F
-****--	3C	4	4
-****-*	3D	2	2
-*****-	3E	AM	4
-******	3F	JUMP	F
*------	40	0	0
-----	41	5	5
-----	42	TIY	A
*----**	43	0	0
-----	44	MA	5
----*	45	AO	1
*---**-	46	CAL	E
*---***	47	ENDS	7
-----	48	JUMP	F
----*	49	4	4
---*-	4A	8	8

program 8 Paper, Rock, Scissors game

Binary LEDs	Address	Command symbols	Command code
-------	00	CH	2
------*	01	AIY	B
-----*-	02	1	1
-----**	03	M-	7
----*--	04	AM	4
----*-*	05	AO	1
----**-	06	CH	2
----***	07	KA	0
---*---	08	JUMP	F
---*--*	09	0	0
---*-*-	0A	0	0
---*-**	0B	AM	4
---**--	0C	AIY	B
---**-*	0D	1	1
---***-	0E	MA	5
---****	0F	AM	4
--*----	10	TIA	8
--*---*	11	D	D
--*--*-	12	M-	7
--*--**	13	JUMP	F
--*-*--	14	0	0
--*-*-*	15	F	F
--*-**-	16	AO	1
--*-***	17	AIY	B
--**---	18	F	F
--**--*	19	M-	7
--**-*-	1A	JUMP	F
--**-**	1B	2	2
--***--	1C	0	0
--***-*	1D	JUMP	F
--****-	1E	2	2
--*****	1F	2	2
-*-----	20	AIA	9
-*----*	21	3	3
-*---*-	22	CIA	C
-*---**	23	0	0
-*--*--	24	JUMP	F
-*--*-*	25	2	2
-*--**-	26	C	C
-*--***	27	CAL	E
-*-*---	28	SHTS	9
-*-*--*	29	JUMP	F
-*-*-*-	2A	0	0
-*-*-**	2B	0	0
-*-**--	2C	CIA	C
-*-**-*	2D	1	1
-*-***-	2E	JUMP	F
-*-****	2F	3	3
-**----	30	6	6
-**---*	31	CAL	E
-**--*-	32	ERRS	8
-**--**	33	JUMP	F
-**-*--	34	0	0
-**-*-*	35	0	0
-**-**-	36	CAL	E
-**-***	37	ENDS	7
-***---	38	JUMP	F
-***--*	39	0	0
-***-*-	3A	0	0

program 9 Sequencer music

Binary LEDs	Address	Command symbols	Command code
-------	00	TIY	A
------*	01	0	0
-----*-	02	TIA	8
-----**	03	1	1
----*--	04	M+	6
----*-*	05	AM	4
----**-	06	CIA	C
----***	07	3	3
---*---	08	JUMP	F
---*--*	09	1	1
---*-*-	0A	5	5
---*-**	0B	TIA	8
---**--	0C	0	0
---**-*	0D	AM	4
---***-	0E	TIA	8
---****	0F	9	9
--*----	10	CAL	E
--*---*	11	SUND	B
--*--*-	12	JUMP	F
--*--**	13	0	0
--*-*--	14	0	0
--*-*-*	15	TIY	A
--*-**-	16	1	1
--*-***	17	TIA	8
--**---	18	1	1
--**--*	19	M+	6
--**-*-	1A	AM	4
--**-**	1B	CIA	C
--***--	1C	7	7
--***-*	1D	JUMP	F
--****-	1E	2	2
--*****	1F	8	8
-*-----	20	TIA	8
-*----*	21	0	0
-*---*-	22	AM	4
-*---**	23	TIA	8
-*--*--	24	1	1
-*--*-*	25	JUMP	F
-*--**-	26	1	1
-*--***	27	0	0
-*-*---	28	TIA	8
-*-*--*	29	4	4
-*-*-*-	2A	JUMP	F
-*-*-**	2B	1	1
-*-**--	2C	0	0

program 10 Gunfighting game

Game results obtained by looking at lit LEDs (tentative name)

When the person on side 4 wins:
○●●●●●●●

When the person on side 7 wins:
●●●●●●●○

When the person on side 4 has a false start:
●○●●●●●●

When the person on side 7 has a false start:
●●●●●●○●

Binary LEDs	Address	Command symbols	Command code
-------	00	TIA	8
------*	01	F	F
-----*-	02	CAL	E
-----**	03	TIMR	C
----*--	04	CH	2
----*-*	05	KA	0
----**-	06	JUMP	F
----***	07	0	0
---*---	08	C	C
---*--*	09	JUMP	F
---*-*-	0A	3	3
---*-**	0B	6	6
---**--	0C	TIA	8
---**-*	0D	3	3
---***-	0E	CAL	E
---****	0F	TIMR	C
--*----	10	CH	2
--*---*	11	AIA	9
--*--*-	12	F	F
--*--**	13	JUMP	F
--*-*--	14	0	0
--*-*-*	15	4	4
--*-**-	16	TIY	A
--*-***	17	3	3
--**---	18	CAL	E
--**--*	19	SETR	1
--**-*-	1A	TIY	A
--**-**	1B	6	6
--***--	1C	KA	0
--***-*	1D	JUMP	F
--****-	1E	1	1
--*****	1F	C	C
-*-----	20	CIA	C
-*----*	21	4	4
-*---*-	22	JUMP	F
-*---**	23	2	2
-*--*--	24	8	8
-*--*-*	25	JUMP	F
-*--**-	26	2	2
-*--***	27	F	F
-*-*---	28	CIA	C
-*-*--*	29	7	7
-*-*-*-	2A	JUMP	F
-*-*-**	2B	1	1
-*-**--	2C	C	C
-*-**-*	2D	TIY	A
-*-***-	2E	0	0
-*-****	2F	CAL	E
-**----	30	SETR	1
-**---*	31	CAL	E
-**--*-	32	ENDS	7
-**--**	33	JUMP	F
-**-*--	34	3	3
-**-*-*	35	3	3
-**-**-	36	TIY	A
-**-***	37	5	5
-***---	38	CIA	C
-***--*	39	4	4
-***-*-	3A	JUMP	F
-***-**	3B	4	4
-****--	3C	0	0
-****-*	3D	JUMP	F
-*****-	3E	4	4
-******	3F	7	7
*------	40	CIA	C
-----	41	7	7
-----	42	JUMP	F
*----**	43	0	0
-----	44	5	5
----*	45	TIY	A
*---**-	46	1	1
*---***	47	CAL	E
-----	48	SETR	1
----*	49	CAL	E
---*-	4A	ERRS	8
---**	4B	JUMP	F
*--**--	4C	4	4
*--**-*	4D	B	B