



SMART MOTOR DEVICES

<https://smd.ee>

**PROGRAMMABLE STEP MOTOR
CONTROLLER SMSD-4.2LAN and SMSD-8.0LAN**

Program examples
Ver. 01

2020



1.Speed parameters: acceleration, deceleration, minimum and maximum speed. Incremental and absolute displacement.	3
2.Smooth and abrupt motor stop.....	4
3. SET_MODE command.....	4
4.Pause.....	7
5.Cycle with counter.....	7
6.Unconditional branching.....	8
7.The implementation of an infinite loop using an unconditional branching.....	10
8.Conditional branching within the same memory block.....	11
9.Conditional branching to other memory blocks. Fullstep speed.	12
10. Signal waiting.....	13
11.Call subprogram.....	14
12. Turning the relay on and off.....	15
13. Search and use of "zero" position and label V.....	16
14. Motor control parameters setting, current and voltage control modes, deenergizing phases.	17
15. Search of "V" position, cycle, relay, zero position	19



1. Speed parameters: acceleration, deceleration, minimum and maximum speed. Incremental and absolute displacement.

It is recommended to set speed parameters at the beginning of the program or before movement commands: acceleration, deceleration, minimum and maximum speeds.

The value of acceleration and deceleration should be in the range from 15 to 59000 steps/sec². The minimum speed value lies in the range from 0 to 950 steps/sec, and the maximum speed should take values from 16 to 15600 steps/sec.

All speed parameters for SMSD-LAN controllers are set as full steps per second, regardless of set microstepping mode.

Increment is an operation in many programming languages that increments a variable. In this case, the number of steps taken by the rotor of the stepper motor increases. The number of rotor steps in this program lies in the range from 0 to 10²² steps. The zero position is the one in which the rotor was located at the time of power switching on, or the position set by command, or the position set with the sensor.

Absolute movement - moving the rotor to the position specified numerically in the program, in other words it is a coordinate.

All movement commands are set as microstepping displacement.

LanProgram1

SET_MIN_SPEED	-> 10 //set minimum speed
SET_ACC	-> 15 // set acceleration
SET_MAX_SPEED	-> 300 // maximum speed
SET_DEC	-> 50 // deceleration
MOVE_F	-> 10000 // forward incremental displacement
GO_TO_F	-> 20000 // forward absolute displacement to the position 20000
GO_TO_R	-> 20000 // backward absolute displacement – the rotor will not move as it is already located at this position.
MOVE_R	-> 20000 // backward incremental displacement
GO_TO_F	-> 20000 // forward absolute displacement to the position 20000 steps.

2. Smooth and abrupt motor stop

When controlling the drive, situations often arise when it is necessary to stop the motor smoothly or sharply. To do this, special functions are provided that, together with a stop, turn off the phase power. They are written as follows:

SOFT_HI_Z -> 0 // Soft stop and removing power from the phases

HARD_HI_Z -> 0 // Hard stop with removing power from the phases

SOFT_STOP -> 0 // Soft stop without removing power from the phases

HARD_STOP -> 0 // Sudden stop without removing power from the phases



3. SET_MODE command

To change the control parameters, such as the phase control method, microstepping mode, holding current, operating current or motor type, the SET_MODE command is used. This command is recommended to start each program. This command can be only applied when motor phases are deenergized.

Phase control methods:

Current mode - when controlling the motor, the maximum current supplied to the motor phase is controlled. To control in this mode, you can use any stepper motor with setting the maximum current in the controller settings. The mode is characterized by high torque, high rotation speed, but it is limited with the maximum microstepping value 1/16;

Volt mode - characterized by a smooth ride and the possibility of microstepping up to 1/128; however, speed and torque are lower compared to the current control mode. This mode is applicable only to those motor drives whose parameter list is stored in the controller's memory.

In order to use SET_MODE command it is necessary to deenergize the phases of the motor before applying this command.

Чтобы менять параметры управления, такие как, способ управления фазами, микрошаг, ток удержания, рабочий ток и тип двигателя, используется функция SET_MODE. Данной функцией рекомендуется начинать каждую программу. Команда SET_MODE может применяться только при обесточенных фазах двигателя, поэтому перед использованием команды SET_MODE рекомендуется добавлять команды HARD_HI_Z или SOFT_HI_Z.

LanProgram3

SET_MIN_SPEED	-> 10 // Minimum speed
SET_ACC	-> 15 // Acceleration
SET_MAX_SPEED	->300 // Maximum speed
SET_DEC	-> 50 // Deceleration
HARD_HI_Z	-> 0 // Sudden stop and deenergizing phases
SET_MODE	-> 160259 // Setting parameters (as in the figure 1)
MOVE_F	->5000 // Forward incremental movement
SOFT_HI_Z	-> 0 // Smooth stop and deenergizing phases
SET_MODE	-> 160690 //Setting parameters (as in the figure 2)
MOVE_R	-> 15000 // Backward incremental movement
SOFT_HI_Z	-> 0 // Smooth stop and deenergizing phases
SET_MODE	-> 159747 //Setting parameters (as in the figure 3)
MOVE_R	-> 200 //Перемещение на 200 шагов



SMART MOTOR DEVICES

<http://smd.ee>

SMART MOTOR DEVICES
SMC PROGRAM
VER.6.0.3

Microstepping: 1/16 Holding current: 50%
Rated current: 2,8A 1) FL42STH33-1334 1.8 deg

Phases control ☒ Current , ☐ Voltage control

Add command

Fig. 1 – control parameters setting

SMART MOTOR DEVICES
SMC PROGRAM
VER.6.0.3

Microstepping: 1/128 Holding current: 50%
Rated current: 2,8A 25) FL57STH76-2804 1.8 deg

Phases control ☐ Current , ☒ Voltage control

Add command

Fig. 2 – control parameters setting



SMART MOTOR DEVICES
SMC PROGRAM
VER.6.0.3

Microstepping: 1 Holding current: 50%

Rated current: 2,8A 1) FL42STH33-1334 1.8 deg

Phases control ☒ Current , ☐ Voltage control

Add command

Fig. 3 – control parameters setting



4. Pause

LanProgram4

SET_MIN_SPEED	-> 10 // Minimum speed
SET_ACC	-> 15 // Acceleration
SET_MAX_SPEED	-> 300 // Maximum speed
SET_DEC	-> 50 // Deceleration
HARD_HI_Z	-> 0 // Sudden stop and deenergizing phases
SET_MODE	-> 160259 // Parameters setting
MOVE_R	-> 2000// Backward incremental movement
SET_WAIT	-> 500 // Pause 0.5 sec = 500 ms
MOVE_R	-> 2000// Backward incremental movement
SET_WAIT	-> 50000 //Pause 50 sec = 50000 ms
MOVE_R	-> 2000//Backward incremental movement

During the program, the motor executes three displacement through a pause.

5. Cycle with counter

Цикл – это повторяющиеся с течением времени и в определенной последовательности действия.

В программе SMC-program-LAN цикл можно задать с двумя параметрами: количество команд в цикле и количество повторений этого цикла.

LanProgram5.0		LanProgram5.1
LOOP_PROGRAM	-> 4098 // Cycle – 2 commands and 4 repetitions	LOOP_PROGRAM
MOVE_R	-> 2000 //Displacement	MOVE_R
SET_WAIT	-> 500 // Pause	SET_WAIT
		MOVE_R



SET_WAIT

The above two examples are identical in execution, but different in assembling.

6. Unconditional branching

Unconditional branching – jumping to a point of a program without checking of any conditions.

Drive control programs in SMC-program-LAN can be written in several tables (Figure 4), each of which is maximum 256 lines long. The only parameter of the unconditional transition will be the serial number of the code line. In SMC-program-LAN you need to specify the program and line numbers in this program, then the program will calculate the line number itself.

SMART MOTOR DEVICES SMC PROGRAM VER.6.0.3

Controller programming - load program mode

Info Additional Settings RS-232 COM4; 8bits; 1 stop bits; parity: None; 115200baud ; Scale: 1

COM4 RS-232

N°	Command	Scaled	Param.	Code
0	Min speed	10	10	SET_MIN_SPEED
1	Max speed	300	300	SET_MAX_SPEED
2	Acceleration	15	15	SET_ACC
3	Deceleration	50	50	SET_DEC
4	Go to program	P:0; C:6	6	GOTO_PROGRAM
5	Move backward	2000	2000	MOVE_R
6	Pause	500	500	SET_WAIT
7	Move backward	2000	2000	MOVE_R

Speed parameters:

- Set MIN speed: 10
- Set MAX speed: 300
- Full step speed: 250
- Acceleration: 15
- Deceleration: 50

Motion control:

- Permanent motion (speed 200) [Forward/Backward]
- Move 2000 [Inc/Abs]
- Shortest way to 4194303 [Pause 500 ms]
- Zero position: [Go to Zero/Set zero/Search Zero]
- Label "V": [Go to "V"/Search "V" (speed 1000)]
- Move till signal to input 0 [Wait till signal to input 1]

Additional commands:

- Call prog. [Go to: Prog.# 0 cmd.# 6] [if input 0]
- Loop program: 2 commands 2 times [Switch to pulse control mode]
- [Return] [Relay ON/OFF]

Stop: [Soft stop] [Hard stop]

Inputs: SM 0 1 2 3 4 5 6 7

Phases energized: [Phases energized] [Deenergize phases]

Status:

- phases energized
- command error
- SW_F
- SW_EVN
- busy
- standby
- Motion: Current speed: 0
- forward
- backward
- motor stall
- constant speed run
- acceleration
- deceleration

Position: Total steps: 0

Fig. 4 – SMC-program-LAN – program tables



LanProgram6

0 SET_MIN_SPEED	-> 10 // Minimum speed
1 SET_ACC	-> 15 // Acceleration
2 SET_MAX_SPEED	-> 300 // Maximum speed
3 SET_DEC	-> 50 // Deceleration
4 GOTO_PROGRAM	-> 6 // Unconditional jump to line 6
5 MOVE_R	-> 2000 // Displacement – this command will not be executed.
6 SET_WAIT	-> 500 // Pause
7 MOVE_R	-> 2000 // Displacement

The lines numbering is added for a better perception in the listing above.

As a result of this program, the rotor of the motor moves only 2000 steps, because line number 5 is skipped.

LanProgram6.0

0 SET_MIN_SPEED	-> 10 // min. speed	0 SET_MIN_SPEED
1 SET_ACC	-> 10 // acceleration	1 SET_ACC
2 SET_MAX_SPEED	-> 300 //max speed	2 SET_MAX_SPEED
3 SET_DEC	-> 50 // deceleration	3 SET_DEC
4 MOVE_R	-> 2000 //displacement	4 GOTO_PROGRAM
5 GOTO_PROGRAM	->256 //unconditional jump to line 0 of program in mem. 0	
6 SET_WAIT	-> 500 //Pause	
7 MOVE_R	-> 2000 // Displacement	

In this example, an unconditional jump is used to set the drive speed parameters recorded by a separate program.

The program from the listing LanProgram6.0 should be written to memory 0, the program from the listing LanProgram6.1 should be written to memory 1 of the controller.

7. The implementation of an infinite loop using an unconditional branching

Control parameters:

- Control mode - current;
- Microstepping - 1/2;
- Holding current - 50%.



LanProgram7

HARD_HI_Z	-> 0 //Sudden stop and deenergizing phases
SET_MODE	-> 159875 // Parameters setting
SET_ACC	-> 15 // Acceleration
SET_DEC	->100 // Deceleration
SET_MIN_SPEED	-> 10 //Minimum speed
SET_MAX_SPEED	-> 100 // Maximum speed
MOVE_F	-> 1000 // Forward movement
SET_WAIT	-> 300 // Pause
SET_DEC	->200 // Deceleration
SET_MAX_SPEED	-> 1000 // Maximum speed
SET_ACC	-> 1000 // Acceleration
MOVE_R	-> 1000 // Backward movement
SET_WAIT	-> 300 // Pause
SET_ACC	-> 250 // Acceleration
SET_DEC	->15// Deceleration
MOVE_F	-> 3000 //Forward movement
SET_WAIT	-> 500 // Pause
GO_TO_R	-> 1000 // Backward movement to coordinate 1000
GO_TO_F	-> 1000 // Forward movement to coordinate 1000. The motor will not move as it is already in this position.
MOVE_F	-> 1000 // Forward incremental movement
GOTO_PROGRAM.	-> 2 // Unconditional jump to the beginning of the program.

In real conditions, when the motor is running with an inertial load, it is necessary to set the smooth deceleration, since an abrupt stop will cause a big value of back emf, which can damage the controller. Connection of external braking resistor allows to absorb some back EMF energy.

8. Conditional branching within the same memory block

Conditional jump is a jump to a specified point of the program, which is done when some condition is true. For SMSD-LAN controllers the condition is a signal to one of inputs. After program executing begins, the controller starts to check inputs. If input signal arrives, the program execution jumps to the line, specified by conditional branching command.

LanProgram8

0 SET_ACC	-> 15 // Acceleration
1 SET_DEC	->100 // Deceleration
2 SET_MIN_SPEED	-> 10 // Minimum speed
3 SET_MAX_SPEED	-> 100 // Maximum speed
4 HARD_HI_Z	-> 0 // Sudden stop and deenergizing phases
5 SET_MODE	-> 151683 //Parameters setting
6 GOTO_PROGRAM_IF_IN1	->10 // Conditional jump - go to line 10 if where is a signal at input IN1



7 MOVE_R	-> 1000 // Backward movement
8 SET_WAIT	-> 1000 // Pause
9 GOTO_PROGRAM	-> 6 // Unconditional jump.
10 SET_WAIT	-> 1000 // Pause
11 MOVE_F	-> 1000 // Forward movement

The program is cycled until there is not a signal at the input IN1. After receiving the signal the program executing goes to line 10, executes the last movement and ends. This is the way to perform an interruption.

9. Conditional branching to other memory blocks. Fullstep speed.

The next program example consists of 3 part, which should be written to 3 different memory blocks (SMSD-LAN controllers have 4 independent memory blocks). Transitions between memory blocks are done on conditions of arriving signals to IN1 or IN0.

Control parameters:

- Control mode - current;
- Microstepping - 1/16;
- Holding current - 50%.

LanProgram9.0 – should written to the memory block 0.

HARD_HI_Z	-> 0 // Sudden stop and deenergizing phases
SET_MODE	-> 160259 //Parameters setting
SET_MASK_EVENT	-> 0 // Signals setting
GOTO_PROGRAM_IF_IN1	-> 256 // Go to command 0 of memory block 1 if there is a signal at input IN1 (LanProgram9.1)
GOTO_PROGRAM_IF_IN0	-> 512 // Go to command 0 of memory block 1 if there is a signal at input IN0 (LanProgram9.2)
SET_WAIT	-> 100 // Pause
GOTO_PROGRAM	-> 0 // If there are not signals go to the program start – command 0 of memory 0..

LanProgram9.1 – should written to the memory block 1.

SET_MIN_SPEED	-> 100 // Minimum speed
SET_MAX_SPEED	-> 100 // Maximum speed
SET_DEC	-> 20 // Deceleration
MOVE_F	-> 1000 // Forward movement
SET_WAIT	-> 1000 // Pause
GOTO_PROGRAM	-> 0 // Go to command 0 of the memory 0 (LanProgram9.0).

LanProgram9.2 – should written to the memory block 2.

SET_MIN_SPEED	-> 200 // Minimum speed
SET_ACC	-> 50 // Acceleration
SET_DEC	-> 50 // Deceleration



SET_MAX_SPEED	-> 1000 //Maximum speed
SET_FS_SPEED	-> 250 // Fullstep speed
MOVE_F	-> 100000 // Forward movement
SET_WAIT	-> 500 // Pause
GOTO_PROGRAM	-> 0 // Go to command 0 of the memory 0 (LanProgram9.0).

Fullstep speed is the speed when the controller turns to a full step mode (regardless of set microstepping mode). Fullstep allows to achieve higher torque and better dynamisc of a motor.

10. Signal waiting

In this example the command “wait till signal” is shown. The program executing is suspended until there is not a signal at an input.

LanProgram10

SET_ACC	-> 100 // Acceleration
SET_DEC	->100 // Deceleration
SET_MIN_SPEED	-> 10 // Minimum speed
SET_MAX_SPEED	-> 100 // Maximum speed
HARD_HI_Z	-> 0 //Sudden stop and deenergizing phases
SET_MODE	-> 160259 //Parameters setting
LOOP_PROGRAM	-> 5123 // Cycle – 5 repetitions of 5 commands
WAIT_IN1	-> 0 // Wait till signal at the input IN1
MOVE_R	-> 2000 //Movement
SET_WAIT	-> 500 //Pause

Every cycle iteration will be executed after receiving a signal at the input IN1. Цикл будет запускаться только по сигналу IN1.

11.Call subprogram

It is possible to call a subprogram from both the current and another memory block. At the end of each subprogram it is needed to add the return command. After the execution of the subprogram is completed, execution of the main program will continue from the moment the command was called.

In this example the subprogram is located in the same memory block as the main program. This example causes an error during executing, as the command RETURN_PROGRAM is called second time without subprogram command.

LanProgram11

0 SET_MIN_SPEED	-> 100
1 SET_ACC	-> 100



2 SET_MAX_SPEED	-> 3000
3 SET_DEC	-> 500
4 MOVE_F	-> 5000 // Executed 1 time
5 CALL_PROGRAM	-> 7
6 LOOP_PROGRAM	-> 2050 // Executed 1 time. Cycle – 2 repetitions of 2 commands.
7 MOVE_R	-> 5000 //Executed 3 times
8 SET_WAIT	-> 500 //Executed 3 times
9 RETURN_PROGRAM	-> 0 // After first subprogram call – returns to the line 5 of the program. The second time it causes an error.
10 MOVE_F	-> 5000 // Will not be executed

The following example shows calling of a subprogram from another memory block.

LanProgram11.0 – should be written to the memory block 0 of the controller.

0 SET_MIN_SPEED	-> 100
1 SET_ACC	-> 100
2 SET_MAX_SPEED	-> 3000
3 SET_DEC	-> 500
4 CALL_PROGRAM	->512
5 MOVE_F	-> 5000 // Executed 1 time
6MOVE_R	-> 5000 // Executed 1 time

LanProgram11.2 – should be written to the memory block 2 of the controller.

0 LOOP_PROGRAM	-> 2050
1 MOVE_F	-> 5000 // This line will be executed 2 times.
2 SET_WAIT	-> 500 // This line will be executed 2 times.
3 RETURN_PROGRAM	-> 0 // return to line 4 of the memory block 0.

12. Turning the relay on and off

In the next program, the relay is turned on by a signal on IN0 and off by a signal IN1

LanProgram12

WAIT_IN0	-> 0 //Wait for signal IN0
SET_RELE	-> 0 // Turn the relay ON
WAIT_IN1	-> 0 //Wait for signal IN1
CLR_RELE	-> 0 //Turn the relay OFF
GOTO_PROGRAM	-> 0 //Go to the start of the program



13. Search and use of "zero" position and label V

When the task is to calibrate the installation with an end sensor, it is convenient to use the "zero search" mode to set the system to its initial position. The "zero search" command is used to start the motor to rotate at a constant speed. Zero position will be accepted when a signal arrives at the SETZERO input.

Label V search works according to a similar pattern. The only difference is that the controller is waiting for a signal from input IN1 and accept the position as "V" position.

In the next program, a zero search is performed first, then, in the opposite direction, a label is searched. Further, being at the mark, the engine performs the specified action, returns to the zero position, which was determined above. The program performs an action, returns to the label, and loops.

In the following example the controller executes searching of zero position, then searching of "V" position in the reversed direction. After that the motor executes a certain displacement and goes back to zero position and then executes another certain displacement. After that the motor goes to "V" position (which was determined before). The last command returns the program to the beginning (cycle).

LanProgram13

SET_MIN_SPEED	-> 100
SET_MAX_SPEED	-> 1000
HARD_HI_Z	-> 0
SET_MODE	-> 160259
SCAN_ZERO_R	-> 100 //search of zero position (backward direction)
SCAN_LABEL_F	-> 1000 //search of "V" position (forward direction)
MOVE_R	-> 1000 //displacement
SET_WAIT	-> 1000 // pause
GO_ZERO	-> 0 // move to zero position
MOVE_F	-> 1000 // displacement
SET_WAIT	-> 1000 // Pause
GO_LABEL	-> 0 // move to "V" position
GOTO_PROGRAM	-> 6 // Unconditional branching

14. Motor control parameters setting, current and voltage control modes, deenergizing phases.

The following example demonstrates the difference between current and voltage control modes. The motor starts from a zero position, then the first subprogram is called (LanProgram14.1) – the motor stops, the phases are deenergized, then the command set_mode changes the motor control parameters according fig.5.



SMART MOTOR DEVICES
SMC PROGRAM
VER.6.0.3

Microstepping: 1 Holding current: 50%
Rated current: 2.8A 1) FL42STH33-1334 1.8 deg

Phase control: ☒ Current , ☐ Voltage control

Add command

Fig. 5 – Control parameters settings

After command RETURN the program executing continues from LanProgram14.0. The motor moves and some vibrations are clearly noticeable.

After the first motion the second subprogram is called (LanProgram14.2) – motor phases are deenergized, then the command set_mode changes the motor control parameters according fig.6.

SMART MOTOR DEVICES
SMC PROGRAM
VER.6.0.3

Microstepping: 1/128 Holding current: 50%
Rated current: 0,1A 19) FL57STH51-2804 1.8 deg

Phases control ☐ Current , ☒ Voltage control

Add command

Fig. 6 – Control parameters settings

The next motion is much smooth and silent.



Attention! Then the voltage control mode is used, it is necessary to point the correct motor model, which is connected to the controller (the model must be selected from the drop-down list in the program).

There is an unconditional brunching command in the end of the main program (LanProgram14.0) – the program executing returns to the first line.

LanProgram14.0

SET_MIN_SPEED	-> 10//
SET_MAX_SPEED	-> 30 //
SET_DEC	-> 20 //
SET_ACC	-> 15 //
HARD_HI_Z	-> 0 //
CALL_PROGRAM	-> 256 // Call subprogram – command 0, memory block 1.
MOVE_F	-> 150 // motion – vibrations are noticeable
SET_WAIT	-> 500 // pause
CALL_PROGRAM	-> 512 // Call subprogram – command 0, memory block 2.
MOVE_F	-> 10000// Smooth motion
SET_WAIT	-> 500// Pause
GOTO_PROGRAM	-> 0 // Unconditional brunching – return to the beginning

LanProgram14.1

HARD_HI_Z	-> 0 // deenergize phases
SET_MODE	-> 159747 // control parameters setting
RETURN_PROGRAM	-> 0 // Return to the main program

LanProgram14.2

HARD_HI_Z	-> 0 // deenergize phases
SET_MODE	-> 133030// deenergize phases
RETURN_PROGRAM	-> 0 // Return to the main program

Attention! Every subprogram calling should be followed by RETURN_PROGRAM command. RETURN_PROGRAM command should be called just after subprogram calling. Otherwise the controller turns to error.

15. Search of “V” position, cycle, relay, zero position

The speed of searching “V” position should be in the range from minimum to maximum speeds. So the suitable values of minimum and maximum speeds are set at the beginning of the following example. During searching a “V” position the motor rotates until a signal comes. The controller stores the position as “V” position after receiving the signal. Then the controller executes some actions and then goes to the zero position. After that the controller turns on the relay and cycles.

For cycle it is necessary to set commands quantity and repetition quantity. If there is a pause command in the beginning of the cycle, the each repetition of the cycle starts just as per this signal.



The next step of the following example – turn the relay of, go to “V” position and repeat commands.

LanProgram15.0

```
SET_MIN_SPEED    -> 500
SET_MAX_SPEED    -> 520
HARD_HI_Z        -> 0
SET_MODE         -> 159875
SCAN_LABEL_F     -> 500
GOTO_PROGRAM     -> 256 // Unconditional branching - LanProgram15.1
```

LanProgram15.1

```
SET_MAX_SPEED    -> 900
SET_DEC          -> 20
SET_MIN_SPEED    -> 850
SET_WAIT         -> 1000 // Pause
MOVE_F           -> 1000 // Movement
SET_RELE         -> 0 // turn the relay on
GO_ZERO          -> 0 // go to zero position
GOTO_PROGRAM     -> 512 // Unconditional branching - LanProgram15.2
```

LanProgram15.2

```
SET_MAX_SPEED    -> 700
SET_DEC          -> 20
SET_MIN_SPEED    -> 500
LOOP_PROGRAM     -> 2050 // cycle – 2 commands, 2 repetitions
MOVE_F           -> 1000 // Movement
MOVE_R           -> 1000 // Movement
GO_LABEL         -> 0 // Go to “V” position
CLR_RELE         -> 0 // turn the relay off
GOTO_PROGRAM     -> 256 // Unconditional branching
```

For this example it is necessary to write the listing from LanProgram15.1 to the memory block 1, LanProgram15.2 to the memory block 2