HDSC 华大半导体 HUADA SEMICONDUCTOR

32 -bit microcontroller

Flash serial programming (TypeB protocol)

Suitable

series	model	series	model
HC32L07	HC32L072PATA HC32L072KATA HC32L072JATA HC32L072FAUA HC32L073PATA HC32L073KATA HC32L073JATA	**HC32F07**	HC32F072PATA HC32F072KATA HC32F072JATA HC32F072FAUA
HC32L17	HC32L176PATA HC32L176MATA HC32L176KATA HC32L176JATA HC32L170JATA HC32L170FAUA	**HC32F17**	HC32F176PATA HC32F176MATA HC32F176KATA HC32F176JATA HC32F170LATA HC32F170JATA HC32F170FAUA
HC32L19	HC32L196PCTA HC32L196MCTA HC32L196KCTA HC32L196JCTA HC32L190JCTA HC32L190FCUA	**HC32F19**	HC32F196PCTA HC32F196MCTA HC32F196KCTA HC32F196JCTA HC32F190JCTA HC32F190FCUA

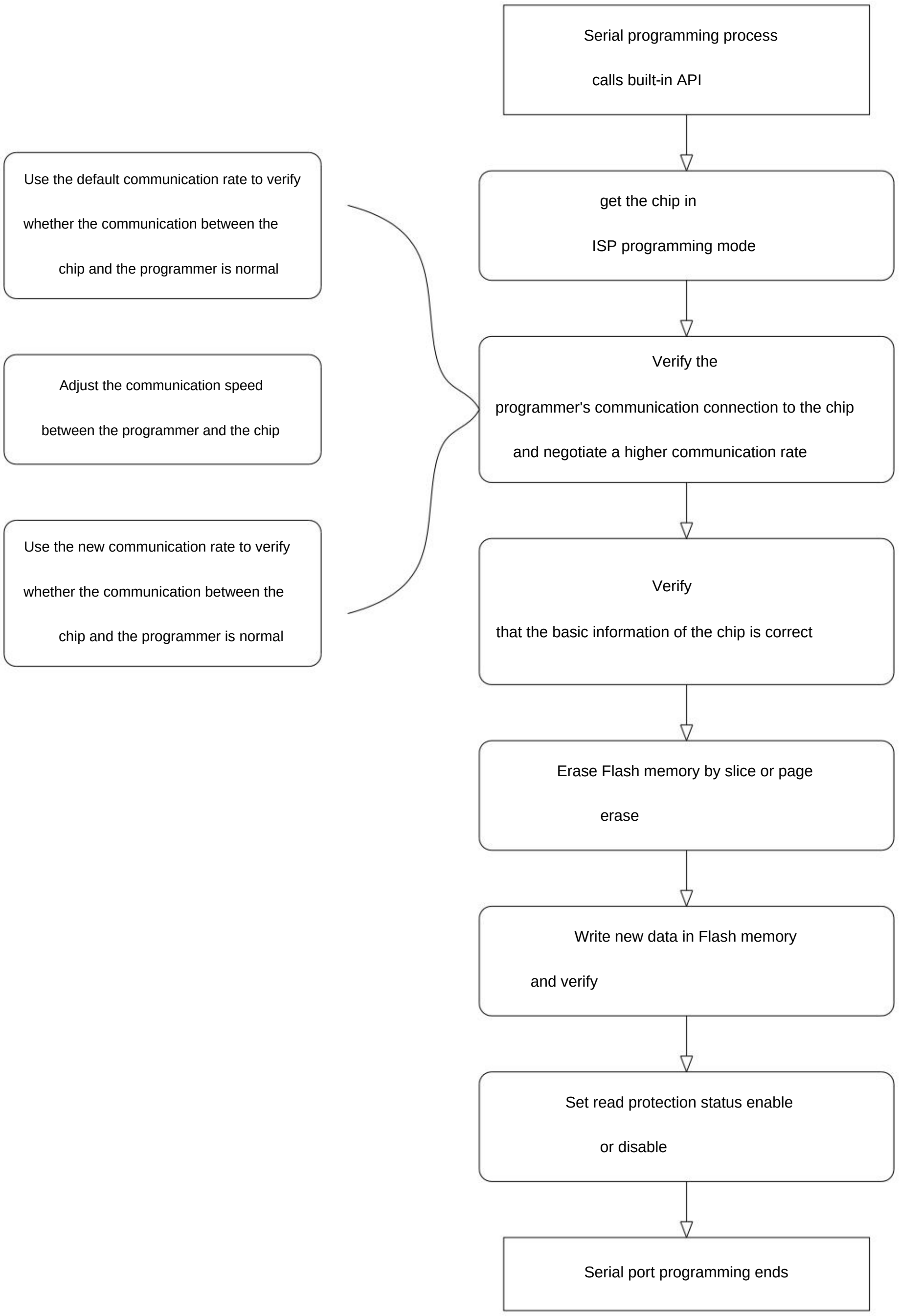
content

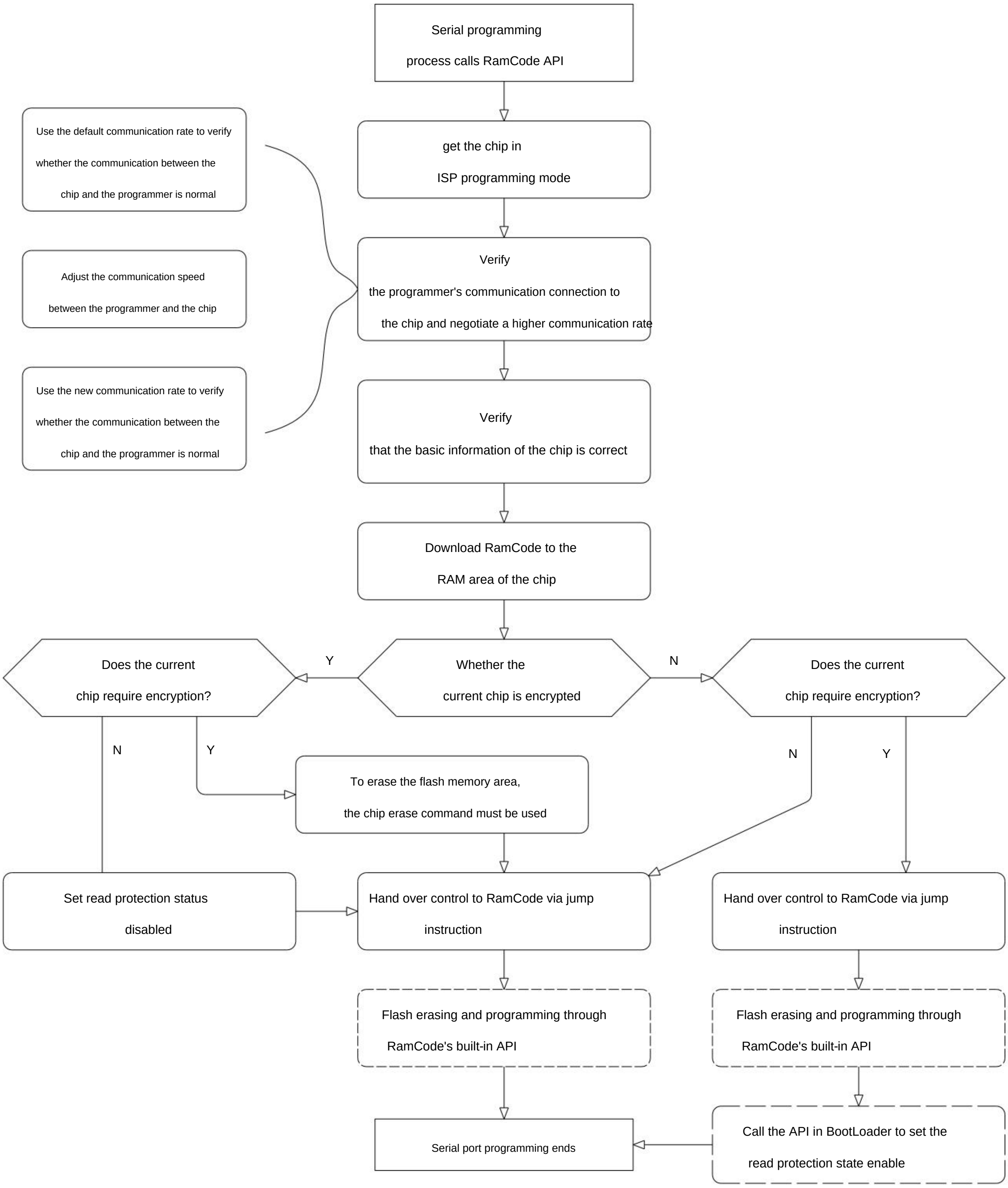
1. Chip programming process

There are 2 kinds of chip programming flow: fully call BootLoader built-in API, mixed call BootLoader and RamCode

within the API.

The specific programming flow chart is shown below.


Serial programming process
calls built-in API
get the chip in
ISP programming mode
Verify the
programmer's communication connection to the chip
and negotiate a higher communication rate
Use the default communication rate to verify
whether the communication between the
chip and the programmer is normal
Adjust the communication speed
between the programmer and the chip
Use the new communication rate to verify
whether the communication between the
chip and the programmer is normal
Verify
that the basic information of the chip is correct
Erase Flash memory by slice or page
erase
Write new data in Flash memory
and verify
Set read protection status enable
or disable
Serial port programming ends


Serial programming process calls RamCode API
get the chip in ISP programming mode
Verify the programmer's communication connection to the chip and negotiate a higher communication rate
Use the default communication rate to verify whether the communication between the chip and the programmer is normal
Adjust the communication speed between the programmer and the chip
Use the new communication rate to verify whether the communication between the chip and the programmer is normal
Verify that the basic information of the chip is correct
Download RamCode to the RAM area of the chip
Whether the current chip is encrypted
Y
N
Does the current chip require encryption?
Does the current chip require encryption?
N
Y
N
Y
To erase the flash memory area, the chip erase command must be used
Set read protection status disabled
Hand over control to RamCode via jump instruction
Hand over control to RamCode via jump instruction
Flash erasing and programming through RamCode's built-in API
Flash erasing and programming through RamCode's built-in API
Call the API in BootLoader to set the read protection state enable
Serial port programming ends

2. The method to make the target MCU enter ISP mode

Step1. The programmer pulls down the RST pin of the MCU;

Step2. The programmer provides a high level to the BOOT pin of the MCU;

Step3. The programmer pulls up the RST pin of the MCU, and the delay is not less than 50ms.

Step4. The MCU enters the ISP programming mode.

3. UART communication format

After the target MCU enters the ISP programming mode, the communication format between the programmer and the chip is shown below.

3.1 Physical layer

Use UART for communication, the initial communication parameters of UART are: 115200 - 8 - N - 1.

The communication rate can be modified by the PPS instruction.

3.2 Protocol layer

The data packet format for uplink and downlink is Head - Len - Info - CRC

Head is the frame header, and its value is fixed at 0x65

The value of Len represents how many bytes Info has, and the value range is 0-255

Info is the actual application layer data, and its length is Len. When going up, the first byte of Info represents the execution status

CRC is the calculated value of all the data of Head - Len - Info, the low byte is sent first

Example of sending and receiving data:

Down: 65 01 10 65F3

Up: 65 09 00 1800080001010600 BA2B

Among them, yellow is Head, blue is Len, green is Info, and gray is CRC.

4. Application layer instructions

command name	Instruction format	Instruction function description
Query	10	The programmer queries whether the target MCU can communicate normally.
PPS	11 DIVN	The programmer negotiates the communication rate with the target MCU.
SetBaseAddr	27 BaseAddr	Sets the base address for read, write, and erase operations.
ChipErase	20	Erase all data in the target MCU flash.
SectorErase	21 Offset	Erase the page where the BaseAddr + Offset address is located.
Blank Check	twenty two	Check whether the data in the flash memory of the target MCU is all FF.
Write Data	28 Offset D1-Dn	Write N bytes of data at BaseAddr + Offset, and verify that the read and Whether the written data is the same. N = 1 - 248.
Read Data	29 Offset N	From BaseAddr + Offset, read N bytes. N = 1 - 255.
ReadOutProtection 2B RdEn		Set or read the read data status of the chip.
Jump	30 Addr	Jump to the address specified by Addr to start program execution.

4.1 Query command

ÿ Command function: The programmer queries whether the target MCU can communicate normally.

ÿ Downstream format: 10

ÿ Upstream format: Status word + HCLK + PRSC + BootLoaderId

ÿ Instruction description: HCLK is 2 bytes, the low byte is sent first, and the unit is MHz.

PRSC is 2 bytes, low byte first, UART clock = HCLK / PRSC.

BootLoaderId is 4 bytes, the low byte is sent first.

4.2 PPS command

ÿ Command function: the programmer negotiates the communication speed with the target MCU.

ÿ Downstream format: 11 DIVN

ÿ Upstream format: status word

ÿ Instruction description: DIVN is 2 bytes, the low byte is sent first.

After the MCU returns the status word, adjust the communication baud rate to HCLK/RPSC/DIVN.

4.3 SetBaseAddr command

ÿ Instruction function: set the base address of read, write and erase operations.

ÿ Downstream format: 27 BaseAddr

ÿ Upstream format: status word

ÿ Instruction description: BasedAddr is 4 bytes, the low byte is sent first.

0x000xxxxx represents the flash memory area.

0x2000xxxx represents the RAM area.

4.4 ChipErase Instruction

ÿ Command function: Erase all data in the target MCU flash memory.

ÿ Downstream format: 20

ÿ Upstream format: status word

ÿ Instruction description: After erasing is completed, all data in the flash memory will become 0xFF.

4.5 SectorErase directive

ÿ Instruction function: erase the page where BaseAddr + Offset address is located

ÿ Downstream format: 21 Offset

ÿ Upstream format: status word

ÿ Instruction description: Offset is 2 bytes, the low byte is sent first.

4.6 Blank Check Command

ÿ Instruction function: Check whether the data in the flash memory of the target MCU is all FF.

ÿ Downstream format: 22

ÿ Upstream format: status word

ÿ Instruction description: None.

4.7 Write Data Command

ÿ Command function: write N bytes of data to BaseAddr + Offset, and verify whether the read and written data are consistent same.

ÿ Downstream format: 28 Offset D1-Dn

ÿ Upstream format: status word

ÿ Instruction description: Offset is 2 bytes, the low byte is sent first.

N = 1 - 248.

When BaseAddr+Offset and the number of bytes to be programmed are both Word aligned, call Word Program operation, otherwise call the Byte Program operation. Word Program only needs half of Byte Program time.

4.8 Read Data Command

ÿ Command function: read N bytes from BaseAddr + Offset.

ÿ Downstream format: 29 Offset N

ÿ Upstream format: status word + D1-Dn

ÿ Instruction description: Offset is 2 bytes, the low byte is sent first.

N is 1 byte, the value range is 1 - 255.

The commercial number can be obtained by reading 0x00100C60~0x00100C6F.

Read 0x00100C70~0x00100C73 to get the capacity of Flash.

Read 0x00100C74~0x00100C77 to get the Ram capacity.

Read 0x00100C78~0x00100C79 to get the FlashSectorSize.

Read 0x00100C7A~0x00100C7B to get the number of pins.

4.9 ReadOutProtection Command

ÿ Command function: set or read the read data status of the chip.

ÿ Downstream format: 2B RdEn

ÿ Upstream format: Status word + RdState + Cnt

ÿ Instruction description: If RdEn is 0x00, any data in the flash memory cannot be read; the SWD interface is prohibited.

If RdEn is 0xFF, any data in the flash memory can be read; the SWD interface is enabled.

When RdEn is 0x55, the current read protection state is not operated, and only the current protection state is returned.

RdState of 0x00 means that the flash memory is encrypted, and the data in it cannot be read.

If RdState is 0xFF, it means that the flash memory is in decrypted state, and the data in it can be read out.

Cnt represents the remaining times that the read protection status of the chip can be rewritten, up to 64 times.

If the MCU is already in the encrypted state, when receiving the command with RdEn 0xFF, all data in the flash memory will be data erased.

4.10 Jump command

ÿ Instruction function: jump to the address specified by Addr to start the program execution.

ÿ Downstream format: 30 Addr

ÿ Upstream format: status word

ÿ Instruction description: Addr is 4 bytes, the low byte is sent first.

Addr can only be 0x2000xxxx or 0x00000000.

4.11 Description of return status word

0x00, representing successful execution

0x10, it means communication CRC check error, need to resend the command

0x11, it means UART communication error

0x20, which means the instruction is not supported

0x21, which means the parameter is not supported

0x30, means no read permission

0x31, means no write permission

0x32, means no jump permission

0x40, which means Write Data failed

0x41, which means Blank Check failed

0X42, which means verification failed

5. Programming example

5.1 Fully call the built-in API of BooLoader for programming

```
//The following is the BootLoader instruction

OUT 65011065F3

IN 650900180008001201150002F72

OUT 6503110200A481

IN 650100E4E3

OUT 65011065F3

IN 650900180008001201150002F72

OUT 650527000010000D09

IN 650100E4E3

OUT 650429600C10F583

IN 65110048433332 4C313936504354410000000 0F1EA

OUT 650120E6C2

IN 650100E4E3

OUT 650122F4E1

IN 650100E4E3

OUT 650527000000009C9C

IN 650100E4E3

OUT 65F32800000000100204501000097010000A301000000000000000000000000000000000

          0000000000000000000000C50100000000000000000C7010000C9010000CB010000
          CD010000CF010000D1010000D3010000D5010000D7010000D9010000DB010000DD0100
          00DF010000E1010000E3010000E5010000E7010000E9010000EB010000ED010000EF0100
          00F1010000F3010000F5010000F7010000F9010000FB010000FD010000FF0100000102000
          003020000050200000702000009020000 80B517480068802149050143144801601448006814
          49014012480160114800681021884 30F490860802000020F4908602E17

IN 650100E4E3

OUT 65F328F00010200E490860802000020D49086010200A4908600B4800F017F81020084908

          608020000205490860064800F00DF8EAE720200040001D0240FF7FFFFF101D0240141D02
          4050C3000000B5002801D0401EFBE700BD05480022874500D90244044802600448804704
          48004700000000002008ED00E00D020000B901000000F00DF8002801D000BF00BF00BF0
          0BF002000BF00BFFFF79DFF00F002F80120704780B500F002F801BDFEE70746384600F0
          02F8FBE7FEE780B500BF00BF024A11001820ABBEFBE72600020000BF00BF00BF00BF
          FFF7D4FFFEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE76D
          AF
IN 650100E4E3
OUT 653128E001FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7
          FEE7FEE7FEE7FEE7FEE7FEE7FEE7000070479C3B
IN 650100E4E3
OUT 650527000001004485
IN 650100E4E3
OUT 6507280000640000000CC18
IN 650100E4E3
OUT 650429000004BD79
IN 65050064000000D927
OUT 65022B004E79
IN 650300003D3B87
```

5.2 Mixed calling BootLoader and RamCode API for programming

//The following is the **BootLoader** instruction

```
-->65011065F3

<--650900180008001001150059 4B

-->65031103007C98

<--650100E4E3

-->650527000010000D09

<--650100E4E3

-->650429600C10F583

<--6511004843333 24C3139365043544100000000F1EA

-->650527000800205C7B

<--650100E4E3

-->65F3280000B81E0020110800200000000097 0F002072B60448016881F3088803490860034800

4700000008002008ED00E0B90F0020C068016873480162734A8918734A914201D306218171704

780B500F0F5F87048016803220A43026000210970 0168CA06D20FFBD101BD10B504006068802

1C902884205D3674940188021890088 4210D200F0DAF8624801680322914302220A4302600021

626811700168CA06D20F03D0FAE705205749887110BD80B501004A6948688023DB029A4205

D3574B984207D3564B9A4204D20A89C96800F012F901BD05204C49887101BD80B50189494A

91800289C168406800F02FF901BD1CB500213B23

<--650100E4E3

-->65F328F0006A461180C16809684068424C4218521E8023DB029A4202D30520A07102E06A4

600F0D0F84248694609880172694609880 90A41720220A08013BD7CB500220092C1680968334E

0125B580394C22726A46406800F0C4F8002802D00098306073BD257273BD01202B4988803149

314A1278FF2A00D100200872704780B500202E490968884701BD704710B5224C00F08BF900F0

0EF90128F9D118212000083000F09FF92000083000F01AF9A071E16821600021A18000280BD1

2048617A89004158002903D020000830884701E00220A071200000F041F9A08800F05DF9607A0

12805D1A0790028D1D1206A00F081F9607A0FE8

<--650100E4E3

-->65F328E0010A28CBD1A0790028C8D1FA200001401EFDD1BFF34F8F0E480E490860BFF34

F8F00BFFDE7101E002080DAFFFFC11C0F002000024000F6EFFF000A1000000C1000041C0020

FC0B10009C0B1000F80E00200400FA050CED00E04A484B4901604B490160704700E00000494

84749C162474AC262C0230360C162C2628A234360C162C262A2238360C162C262424BC360C1
```

```
62C262414B0361C162C26290234361C162C262B423DB008361C162C2623B4BC361C162C2620
0230362C162C262DB430363C162C2624363C162C2620364C162C2624364704730B5002300240
3E005785B19401C641C8C42F9D31380002030BD30B5238F

<--650100E4E3

-->65F328D0020300002400E0641C8C4208D21D006B1C2D78FF2DF7D000191060012030BD00

2030BDF0B5224B1C4CDC601C4CDC6001241D680326B54325431D6095070BD1150415D01D6
82D092540FBD10D680560091D001D121FF3E7002507E01E6836092640FBD14E5D47193E706
D1C9542F5D3186800092040FBD1F0BD10B5002304E004780C70401C491C5B1C9CB29442F7D
3002010BD00002C0002405A5A0000A5A5000000000240E0A5010040D10C00701700002000024
010B50A000021002303E004780919401C5B1C9CB29442F8D3C8B210BD4848018809290BDB4
7494A78052A09D08A79C97909021143093100888142O4D0EF55

<--650100E4E3

-->65F328C003002070470088O928FAD10120704738B50400002000253B4A118810803A480278

2270427862708278C3781B021343027912041A4343791B06134363608379C27912021A43228163
689B185B1E63612378492B16D163780B2B01DA002B01D1022510E0002A02D002000832E2604
218521E12782274491E89B2FFF7A4FF217C884200D00125280032BD38B504001E4DA0796870
2068A8702068000AE8702068000C28712068000E6871A088A871A088000AE871A188083189B2
2800FFF783FFA1886918087231BD80B50100093189B20E4800F086F801BD38B500F096F80023
09490A88002A01D149280AD1074A0C88074DECE1

<--650100E4E3

-->65F328B004AC4204DA0B885C1C0C80D05431BD13700B8031BD341E0020041C002009020

000002200BF094202D0491E4254FCD1704730B500210022234BD4B2A5005D59A84209D0521C
D4B20C2CF6DB00BF38A0405A1D49086030BD6100F7E770B51B48002101601B491B4ACA61
1B4A4A601B4A4A600A68830D13430B6019491A4ACB681340CB60194B02240725ED435E682
E4026435E600E6832400A60CA68144E1640CE601A6815402C431C60420C0B6813430B600B6A
1A430A629421890001609C214900416170BD980E00201800004004000040042000400100809 25A
5A0000A5A50000000D0240FFBFFFFF340C0240FFDFFFFFC210

<--650100E4E3

-->65F328A00530B500228023DB050AE0045D1C601C69A507ED0FFBD05C690225AC435C61

521C94B28C42F1D330BD8020C0050169C907FCD5416901229143416100 68C0B270473801D00
09C004E0034001A00170027000C000D000300020038B5124812498160124A82600368E024E400
1C430460104B104CDD882540C5609D882540C5601B881C40C4608160826001680B4A0A40026
00620FFF707FEFFF768FFFFF79DFD002032BD00BF002000405A5A0000A5A50000000C1000F
```

```
F070000FFF8FFFF70B40123002413E00168001D194202D04D466D1E49190C60091D121F042A
FAD20D00960701D50C80AD1C1A4000D02C700268001D1F39

<--650100E4E3

-->65F3289006002AE7D170BC70478025000040380000004B00000096000000E1000000C201000
0F40100002C010000E803000084030040420F0060E3160030B401220168001D00290FD00368C31
844680830144202D04D466D1E64191D6825601B1D241D091FECD0F8E730BC70470000000031
080020490800206708002 0AB080020D9080020ED0800202B090020550900206B09002077090020
10B5074979441831064C7C44163404E0081D0A68891888470100A142F8D110BD080000002800
000011FFFFFF34020000041C0020000000006DFFFFFF040000006000000001C002000000000001
20C046002801D0FFF7D4FF00BF00BF6710

<--650100E4E3

-->654B288007002000BF00BFFFF737FF00F000F880B500F002F801BDFEE70746384600F002F8
FBE7000080B500BF00BF024A11001820ABBEFBE72600020000BF00BF00BF00BFFFF7D6FF0
00000009FAC

<--650100E4E3

-->65022B55667C

<--650300FF3C7269

-->650530040800202C8D

<--650100E4E3
```

//The following is the **RAMCODE** instruction

```
-->490400000000000002080200208D0200002B0200005F0200000000000000000000000000000000

-->4901000000000040040420F00DF

<--4900000000000000049

-->4902000000000000004B

<--4900000000000000049

-->49070000000004000004000058

<--4900000000000100014B

-->490400000000000002080200208D0200002B0200005F0200000000000000000000000000000000
00000000000000000000000000001703000000000000000000002303000049030000 4B0300004D0
300004F03000051030000530300005503000057030000590300005B0300005D0300005F03000061
0300006303000065030000670300006903000 06B0300006D0300006F030000710300007303000 07
5030000770300007903000 07B0300007D0300007F030000810300008303000085030000870300 00
```

```
8903000070B505002800451E002811D00020234908608026760422480068FA21890000F04DF830
1A04001D4800688442EAD2FAE770BD80B51B4800688021490501431948016018480068802109
0601431548016015480068202188431349086013480068072188431149086011480068202188430F
4908600F480068202188430D4908600D48FFF7BDFF0B48006820210143094801600948FFF7B4F
FECE7000018E000E00000002020200040CC0D0240D40C0240C00D0240C80D02401027000000
22030A8B420BD203098B4219D243088B422ED2411A00D20146524110467047FF2209023FD01
2068B4205D3121209028B4201D31212090203098B4219D300E0090AC3098B4201D3CB01C01A
524183098B4201D38B01C01A524143098B4201D34B01C01A524103098B4201D30B01C01A524
1C3088B4201D3CB00C01A524183088B4201D334

<--490000000000000049

-->4904000200008A018B00C01A524143088B4201D34B00C01A5241884200D3401A5241CFD2

01461046704708B500F002F808BDFEE7704730B401210268001D002A0FD00368C31844680830
0C4202D04D466D1E64191D6825601B1D241D121FECD0F8E730BC7047FEE7704780B5074800
8880B207490968C90AC9024005400D084303490860FFF7F0FF01BD0000080C10000C20004005
480022874500D902440448026004488047044800470000000002008ED00E063020000390300001
0B5074979441831064C7C44163404E0081D0A68891888470100A142F8D110BD0800000018000
00053FFFFFF04000000600000000000000200000000000F00BF8002801D0FFF7DCFF0020C046C0
46FFF7F6FE00F002F80120704780B500F002F801BDFEE70746384600F002F8FBE7FEE780B5C
046C046024A11001820ABBEFBE726000200C046C046C046C046FFF7D6FF00093D00FEE7FEE
7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7F
EE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE745

<--49000002000000004B

-->49060000000004008A030000E0

<--490000000000002005018B3

-->490900000000000052

<--490000000000000049
```

6. Version Information & Contact Information

date	Version revision record
2019/11/27	Rev1.0 first release
2020/6/29	Rev1.1 updated supported models and corrected some typos

If you have any comments or suggestions in the process of purchasing and using, please feel free to contact us.

Email: mcu@hdsc.com.cn

Website: www.hdsc.com.cn

Mailing address: 10th Floor, Block A, No. 1867, Zhongke Road, Pudong New Area, Shanghai

Postcode: 201203