

MG82G5E32

IHRCO Frequency Fine-

tuning Library

User Guide

Contents

1	IHRCO and its frequency fine-tuning overview	3
2	IHRCO Frequency Fine-tuning Library	4
2.1	void Initial_AdjIHRCO(void) Function Description	4
2.2	void Adjust_IHRCO(signed Char Step) Function Description	5
2.3	void ReInitial_IHRCO(void) Function Description	5
3	Precautions for using this library	5
3.1	Resource of MCU occupied	5
3.2	Other matters needing attention	5
4	Revision History	6

1 IHRCO and its frequency fine-tuning overview

MG82G5E32 MCU has high precision internal High-frequency RC Oscillator(IHRCO), the built-in IHRCO provides to kinds of frequency for user selected. The IHRCO of each MCU before leaving the factory has been adjusted for the above two frequencies, so that the frequency error is controlled within a certain accuracy range(For detailed specifications, please refer to the MG82G5E32 datasheet)

For some specific purposes, users of MG82G5E32 need to be able to dynamically fine-tune the frequency of the built-in IHRCO through the MCU program according to the actual application of the system. In order to satisfy high-end users of such MG82G5E32, the provision of this library will meet the needs of users for this function.

The name of this library is: MG82G5E32_Adj_Freq.LIB. This library is placed in the <Sample Code> folder of this development kit.

2 IHRCO Frequency Fine-tuning Library

This MG82G5E32 IHRCO frequency fine-tuning library (MG82G5E32_Adj_Freq.LIB) provides a total of two functions that can be called:

- (1) void Initial_AdjIHRCO(void)
- (2) void Adjust_IHRCO(signed Char Step)

Its operating principle is explained as follows:

2.1 void Initial_AdjIHRCO(void) Function Description

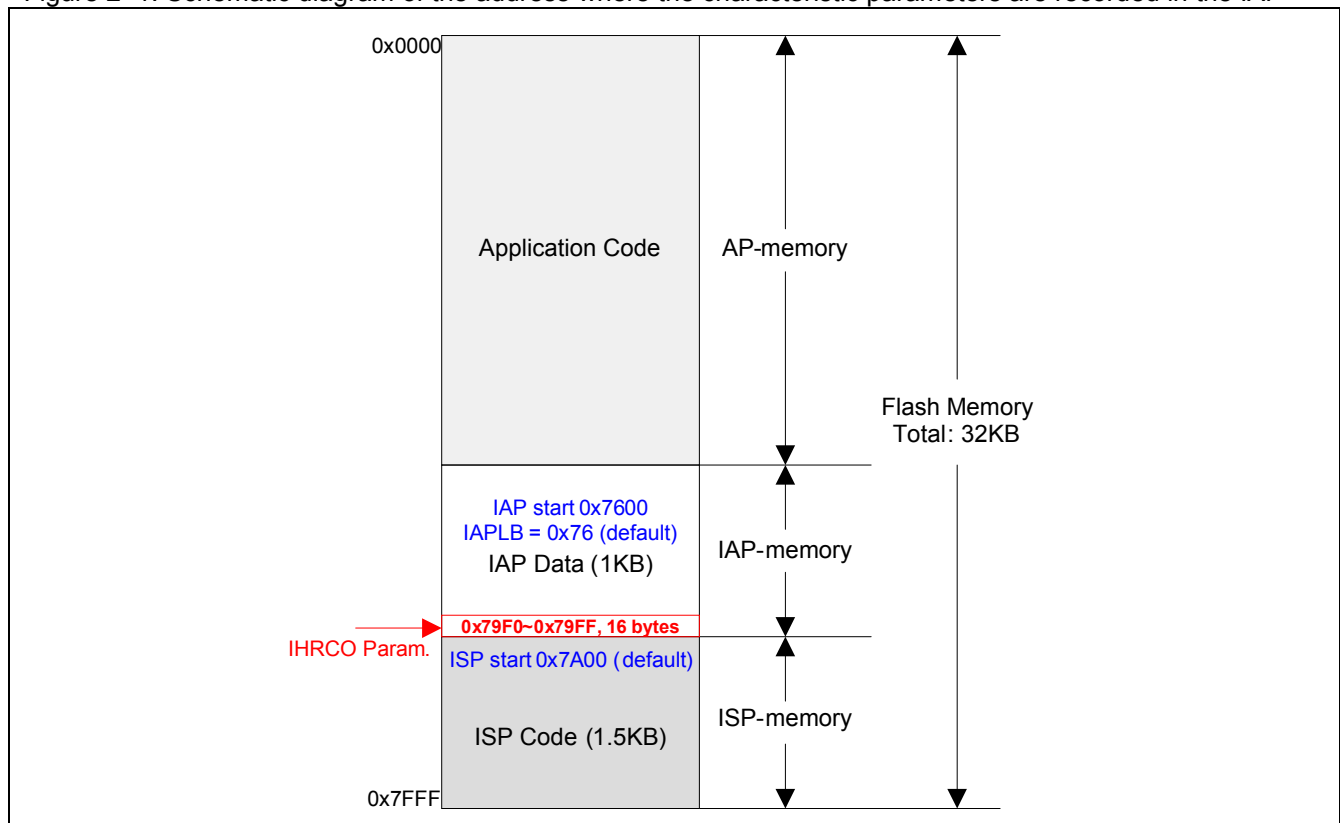
This function is the initialization function for IHRCO frequency fine-tuning. It will calculate the characteristic parameters of the chip IHRCO and record them in the IAP area of the MCU Flash. This characteristic parameter will provide the Adjust_IHRCO function to perform the finest frequency adjustment and calibration function, the operation of this function is as follows:

2.1.1. This function and characteristic parameters are not preset in the 5E32 MCU produced by Megawin, so it is necessary for the user to integrate this function library into the integration when developing the program.

2.1.2. When the user needs to use the IHRCO frequency fine-tuning function, the 5E32 MCU must execute this initialization function at least once to record this characteristic parameter in the MCU IAP.

2.1.3. This feature parameter record requires a total of 16 bytes, and the storage address is the last 16 bytes of the factory default IAP space of MG82G5E32, which is stored in the MCU Flash address 0x79F0 ~ 0x79FF. (For the usage and precautions of IAP, please refer to the MG82G5E32 datasheet)

Figure 2–1. Schematic diagram of the address where the characteristic parameters are recorded in the IAP



2.1.4. If this characteristic parameter already exists in the IAP, the MCU program calls this initialization function, and this function will only check this characteristic parameter to save program running time. Taking the MCU clock as 12MHz as an example, this initialization takes time: 55us (microseconds).

2.1.5. If this characteristic parameter does not exist in the IAP of the MCU, and the 16 byte data of the IAP is all "FF", taking the MCU clock as 12MHz as an example, the initialization will take time: 3.6ms (milliseconds).

2.1.6. If other data is recorded in the IAP of the MCU, taking the MCU clock as 12MHz as an example, the initialization will take 35ms (milliseconds).

2.2 void Adjust_IHRCO(signed Char Step) Function Description

This function is the adjustment function for IHRCO frequency fine-tuning. Users of MG82G5E32 can call this program at any time to perform fine-tuning of IHRCO frequency according to their application. The operation is as follows:

2.2.1. The input parameter Step of the function is a signed number, Step represents the order to be adjusted, and its value range is limited to -30 ~ +30 in decimal. If the step input parameter value exceeds this range, this function will not perform any fine-tuning actions.

2.2.2. After executing this function, the 12MHz frequency of IHRCO will change accordingly. (T1CKO output of the actual measurement sample code)

2.2.3. Taking the MCU clock as 12MHz as an example, it takes about 60~105us to execute this function once.

2.2.4. The difference of IHRCO frequency adjusted by each level of this function is < 0.4%, typically about 0.2%.

2.2.5. IHRCO frequency stabilization time after function execution < 30us. (Please refer to IHRCO Characteristics in MG82G5E32 datasheet)

2.2.6. The frequency range of IHRCO that can be adjusted using this function: -12% ~ +12%. In other words, it can be adjusted up to 60 steps in total.

2.3 void ReInitial_IHRCO(void) Function Description

This function is a recovery function for fine-tuning IHRCO frequency. Users of MG82G5E32 can call this program according to their application conditions to restore IHRCO to the factory default 12MHz. Take the MCU clock as 12MHz as an example, execute this function once. It takes about 9us.

3 Precautions for using this library

3.1 Resource of MCU occupied

3.1.1. The program memory (Flash) occupied by this library < 1.5K byte. (Excluding the 16 byte characteristic parameters in IAP)

3.1.2. The RAM occupied by this library is < 36 byte.

3.1.3. When the Initial_AdjIHRCO function is executed, Timer 3 will be used, but Timer 3 ISR will not be used. After the function is executed, Timer 3 will be released and used by the MCU.

3.2 Other matters needing attention

3.2.1. When implementing Initial_AdjIHRCO, be sure to let the MCU operate in a stable and undisturbed environment to improve the accuracy of its characteristic parameters.

3.2.2. The IAP use of MG82G5E32 has its own characteristics. IAP contains several pages, one page has 512 bytes. If the user has other parameters and IHRCO characteristic parameters stored in the same IAP area, please refer to the MG82G5E32 datasheet for correct use to avoid the loss of the two parameter data.

3.2.3. Perform fine-tuning of the IHRCO frequency, and the CPU clock will also change accordingly. Users should pay attention to the limitation of the CPU clock. (Be sure to refer to the MG82G5E32 datasheet for the specifications of the CPU clock)

3.2.4. This development kit contains a directory of <Sample Code>, in which main.c is the sample program that uses this library.

3.2.5. This library (MG82G5E32_Adj_Freq.LIB) is placed in the <Sample Code> directory of this development kit.

3.2.6. When users develop a new project, they need to store this library in the project development directory of ARM Keil.

4 Revision History

Revision	Description	Date
V1.00	(1) Initial release.	2021/06/01