How to setup different reference clock using Registers 1Ch and 1Dh?

I have developed an Arduino library for BK1088E.

Most of function available in BK1088E device has worked well.

I followed the BK1086-88E Datasheet v1.4. However, some important information is not available in details in that Datasheet. I would like to know if you can share more information that can help me to improve the BK108X Arduino Library.

Some difficulties:

Make the BK1088E work with other reference clock. I could make it work only with 32.768kHz.

Improve the FM/RDS functions by getting error information or setup the system properly.

Currently, the Arduino Library that I developed is working in more than 26 boards or microcontrollers. Including: ESP32, Raspberry PI Pico, ATMega 328, Arduino DUE, Aerduino Mega, STM32 etc.

<https://github.com/pu2clr/BK108X>

1. Programming Guide 3.1 Control Interface Bus The BK1086/88 can be controlled by an MCU using either a 2-wire I2C mode or a 3-wire SPI mode, supporting clock signals of up to 20MHz. The communication mode can be selected by setting the logic level of the MODE pin (PIN 7). When MODE = 0, it operates in I2C mode, and when MODE = 1, it operates in SPI mode.

In I2C mode, SCLK serves as the clock signal, SDIO is the data signal, and SEN is not used. The SEN pin can be left unconnected or driven high.

An I2C read or write operation begins with a Start condition and ends with a Stop condition. After the Start condition, the MCU needs to output an 8-bit Device ID to the SDIO line. The Device ID for BK1086/88 is 0x80.

After outputting the Device ID, the MCU continues to send an 8-bit Control Word to the SDIO line. The Control Word consists of a 7-bit starting register address and a read/write bit (1 for read operation, 0 for write operation). For example, if the starting register address is 0x03 and you need to read data from the device, the ControlWord would be (0x03 << 1 + 1) = 0x07.

Once the Control Word is outputted, you can either write data to the SDIO line (for write operation) or read data from the SDIO line (for read operation).

//inicialização padrão do AM

código UINT16 HW\_Reg\_AM[]=

{

0x8000,

0x1080,

0x0281, //REG2 não é mudo por padrão, desative SEEK (busca automática de hardware), ligue, ative Silenciamento suave, mono não obrigatório, estéreo não obrigatório.

// Nota: AM só tem estações mono, não existe separação. 0x0004,

0x60C0, /REG4/interrupção ativa

0x1FDD, //Padrão AMl 522`1710KHz banda de frequência, passo de 9KHz, volume padrão 0x1D. 0x0930,

0x2901,

0xAC8C,

0x1093,

0x402D,

0x004B,

0x0000,

0x0000,

0x0000,

0x0000,

0x7B11, //16

0x0C00,

0x4000,

0x4344,

0x878E, //20

0x7812,

0x4000,

0x0B55, //23

0x341C, //24

//Se a entrada AM for amplificada por um 9018, por favor, coloque o capacitor de sintonia interno do chip no mínimo, ou seja: REG24=0xb4 1c;

0x000D, 0x0001, 0x48D4, 0x0000, 0x0200, 0x80AA, //30 0x0000, //31 0x0EF7, 0x0600, 0x0000, 0xF438, //35 0x0880, 0x8c06, 0x8400, 0x 0000, 0x4400//40

};

//Inicialização padrão do FM

código UINT16 HW\_Reg[] = {

0x8000, //0

0x1080,

0x0281, //Sem mudo por padrão, desligue SEEK (pesquisa de canal automática de hardware), ligue, habilite o mudo suave, não obrigatório mono, estéreo não obrigatório

0x0000,

0x60C0, //REG4

//Reg4(bit6~8) Ajuste do nível de conversão estéreo/mono O padrão é 011, 000 é o mais baixo e 111 é o mais alto.

//Quando for 000, a saída será sempre estéreo, e o sinalizador ST será sempre 1;

//Quando for 111, a saída será sempre mono e o sinalizador ST será sempre 0. //Quanto maior, mais fácil é resolver em mono. Menor é mais fácil de analisar em estéreo.

//As configurações mono e estéreo precisam ser definidas junto com reg2 (mono/estéreo obrigatório) e bit7~13 (análise estéreo) de Reg17.

0x37DA, //REG5

0x086E,

0x0901,

0x9C90,

0x17A0,

0x402B, //10

0x0040,

0x0000,

0x0000,

0x0000,

0x0000, //15

0x7B11,

0x0800,

FM padrão: banda de frequência de 87~108Mhz, passo de 100Khz, volume padrão 0x1A. BK1086/88

//Reg7(bit7~13) Nível de resolução estéreo. Padrão 0x10 //000 0000 é o menor, o mais fácil de analisar em estéreo

//111 1111 é o maior, o mais fácil de analisar em mono

0x4000,

0x4144,

0x829A, //20

0x7812,

0x43BB,

0x0B41,

0x143C,

0x000E, //25

0x0000,

0x48D4,

0x0000,

0x0200,

0x80AA, //30

0x0000,

0x0EF7,

0x0600,

0x0000,

0x7000，//35

0x0880，

0x8D83，

0x8000，

0x0000，

0x4400//40};

Power saving status

When you don't need to listen to the radio, you can write DISABLE (REG2[6]) and ENABLE (REG2[0]) as 1 by controlling to make BK1086/88 enter the power down state. At this time, the BK chip only consumes a very small current (Typical: 19uA), but the operation of reading and writing registers can still be performed.

When the BK1086/88 needs to recover from the power saving state to the working state, you can write DISABLE (REG2[6]) as 0 and ENABLE (REG2[0]) as 1. Since the state of the registers is still retained when power down, there is no need to reinitialize all the registers, and the BK1086/88 can resume work.

Search stations

BK1086/88 provides two different ways to search channels: hardware search and software search: hardware search is performed by BK1086/88

Seek function to search until a valid station is found or the search fails in the entire frequency band and exits; the software search is

Using the Tune function of BK1086/88, the upper-level software needs to Tune once on each frequency point until it finds a valid

Radio station.

The advantage of hardware search is that it takes up less MCU resources and is faster. The advantage of software search is that it is more flexible. For example, it can be used to display the progress of the search.

Hardware search

Also take FM as an example for hardware search instructions: The user needs to set the following parameters before starting hardware search:

Radio frequency range BAND<1:0> (REG5 [7:6])

Channel search step SPACE<1:0> (REG5 [5:4])

Frequency point setting TUNE(REG3[15]) must be set to '0' search mode (return search at the frequency band boundary or stop searching) SKMODE(REG2[10]) 0: search to the boundary and continue searching

1: Stop searching at the boundary. Search direction (search up or search down) SEEKUP(REG2[9]). 0: Search downward from the current frequency point

1: Search upward from the current frequency point

After setting these parameters, set SEEK (REG2[8]) to 1, BK1086/88 will start Seek. Similar to the Tune operation, when the Seek operation is completed, STC (REG10[14]) will be set to 1, and the GPIO2 pin will generate a 5ms low-level pulse when the STC interrupt is enabled. Read the value of SF/BL(REG10[13]) to judge whether Seek is successful.

After the Seek operation is completed, the user can read the values of REG9, REG10 and REG11 to obtain SNR, RSSI, ST and READCHAN (current frequency point) information.

Software search

Since the software search station uses the Tune function, its settings are the same as Tune. Need to read every time Tune is finished

The value of the register is judged by the upper-layer software whether it is the real one. Then Tune the next frequency point and repeats until the entire frequency band is searched.

Program example

3.2.3.2 Canal de pesquisa de software

Como o software usa a função Tune, suas configurações são as mesmas do Tune. Precisa ler toda vez que o Tune terminar

O valor do registro é julgado pelo software da camada superior, seja verdadeiro ou não. Em seguida, sintonize o próximo ponto de frequência e repita até que toda a banda de frequência seja pesquisada.

Exemplo de programa:

//defina BAND = 11, defina ESPAÇO = 10

//configura SKMODE = 1, SEEKUP = 1

//definir CHAN = 0x00, definir a frequência para 87MHz

// limpa o bit de sintonia

//definir SEEK = 1, iniciar a busca

BK1086/88

Exemplo de programa:

//aguarda busca completa

Passo 1

Mov 0x0A1F, 05h

CAN = 0,

Passo 2

Valor = CHAN | 0x8000

Mov Value, 03h // defina CHAN = 0x00, defina a frequência para 87MHz

Aguarde 50ms

Mov 0x0000, 03h //limpa bit de Tune

Lê 09h, 0Ah, 0Bh //Lê parâmetros como RSSI, SNR e Desvio de Frequência, e julga se é uma estação real Se for uma estação real, memorize READCHAN

CAN = CAN +1

Se a frequência for maior que o limite de banda, vá para a Etapa 3

Valor = CHAN | 0x8000

07. Start

08. Set BAND, SPACE and other parameters

09. Tune to start frequency

10. Wait 10ms

Read STC bit

Seek Tune Completed? STC = 1? NO, go to 10.

Read RSSI, SNR and other status

## Commun questions extracted from Beken

Q1. What is the difference between BK1086 and BK1088 chips?

BK1086 only supports FM and AM, while BK1088 supports SW and LW in addition to FM and AM.

Q2. Can the BK1086/88 chip antenna input end without a DC blocking capacitor?

The input DC operating point of the front-end low-noise amplifier of the BK1086/88 chip has been set to 0V internally. so

When using the ground wire as the antenna input, no DC blocking capacitor is required. If there is a DC blocking capacitor (FM mode>100pF, MW mode>270nF) in the external circuit design, it will not affect the receiving performance.

Q3. Is it necessary to add an LC resonant circuit to the FM input terminal of the BK1086/88 chip?

Adding an LC resonant circuit can better improve the FM receiving sensitivity. If there is a high requirement for this performance, it must be join in.

Q4. Must the SEN and MODE pins of the chip be controlled by IO?

You don't need to. If the control mode is I2C mode, the SEN pin can be connected high or floating. MODE pin can

directly to ground.

Q5. Does the chip SEN, SCLK, SDIO have internal pull-up resistors?

There is already a 40K pull-up resistor inside the chip, so in general, the external pull-up resistor can not be used. In reality

In actual applications, the communication may be caused by the weak driver of the external main control platform or the long wiring of the three control signals.

fail. In these cases, the pull-up resistor value needs to be reduced.

Q6. What are the amplitude and precision requirements of the chip for the external input clock?

The input clock accuracy required by BK1086/88 chip is < 200ppm. When the external input clock frequency is 32.768KHz

When, the required clock signal peak-to-peak value is greater than 750mV. When the external input clock frequency is 24MHz, the peak-to-peak value of the clock signal required by BK1086/88 is greater than 1.35V. The waveform of the clock signal is not limited.

Q7. What requirements do BK1086/88 series chips have for the accuracy of peripheral devices?

BK1086/88 is highly integrated, and has no high-precision requirements for peripheral devices except crystals.

Q8. According to the software operation provided in the software manual, why there is no sound after booting, and the return value of reading ChipID is 0xFF?

If the above phenomenon occurs, it can be judged that the chip has not been effectively initialized and is still in the power down state. Check as follows:

1) Check whether the chip soldering is correct;

2) Check whether the DC voltage of all power supplies and grounds is correct;

3) Check whether the hardware mode configuration of the chip meets the software requirements;

4) Check whether the port configuration of the clock and data pins in the port configuration file is consistent with the actual; 5) Make sure that the initialization function in the program has been executed.

If there are no problems with the above items, the read ChipID should be 0x1080, indicating that the chip has been initialized normally.

Q9. After the initialization is completed normally, there is no corresponding station after setting the frequency, or there is a large deviation between the frequency displayed in the channel list and the actual station frequency after automatic search, how to solve this problem? The above situation can be concluded that the reference clock is not input normally , please check as follows:

1) Make sure the clock input mode selection in the initialization settings is correct

2) If an external clock input is used, use an oscilloscope to directly check RCLK to see if the input clock meets the requirements;

3) If an internal crystal oscillator circuit is used, use an oscilloscope to directly check RCLK to see if it starts to oscillate;

4) The initialization file provided by Beken corresponds to the reference clock of 32.768kHz.

Q10. When using reference clocks with different frequencies, how should the internal registers be set?

Adapt to the clock signal used by setting REG29[0:15], REG28[1:0].

The setting method is as follows: First, according to the formula (Freq/512)+0.5, get an 18-bit hexadecimal number, then set Bit[17:16] to REG28[1:0], and Bit[15:0] to to REG29[0:15].

Note: The bit order of REG29 is opposite to the calculation result, that is, Bit[15] is REG29[0], and Bit[0] is REG29[15].

Example: frequency is 12MHz

12000000/512+0.5=23438, converted into hexadecimal number is 0x5B8E. Converted again to binary is 00 0101 1011 1000 1110. Its high 2 bits are 00; reverse the order of the low 16 bits to get 0111 0001 1101 1010, convert to hexadecimal to get 0x71DA.

So the configuration of the 12MHz clock is changed as follows: REG28 remains 0x0000 unchanged;

REG29 changed from 0x0200 to 0x71DA;

Q11. After listening to the radio station normally, first exit the FM function, and then enter the FM function again, only the background noise can be heard, instead of the desired radio state when exiting last time, why?

This phenomenon can be judged as after the power down, the system has not reached a stable state after the initialization is turned on again. set the frequency. It can be solved by appropriately increasing the delay between initialization and setting frequency.

Q12. Why read and write registers are very unstable, sometimes normal, sometimes fail?

The above situation is generally caused by the long I2C control line or slight interference. The number of pull-up resistors can be reduced value to enhance the pull-up capability.

Q13. Why is the data obtained when evaluating chip FM sensitivity much worse than the data provided by the datasheet?

The sensitivity data provided by the Datasheet is tested in a shielded system to ensure that there is no external interference. Most users do not have this condition when evaluating the sensitivity performance, so the test value will be lower than the nominal value of the datasheet. When evaluating, please pay attention to the following items, which can improve the evaluation performance and get as close as possible to the nominal value of the datasheet.

1) When evaluating in an open environment, please make sure that there are no external radio stations near the tested frequency point (+-500kHZ)

2) Use shielded wires as much as possible for the connecting wires between the signal source and the RF input of the chip. If using a normal cable, please keep the length as short as possible while keeping the interface connection intact.

Q14. When evaluating the BK1086/88, when the input signal amplitude drops to a certain level, the output amplitude suddenly drops a lot, which makes it impossible to evaluate the sensitivity. Why?

In the initialization file provided by BEKEN, the soft-mute function (soft-mute) inside the chip is turned on. the function

When the input signal is reduced and the input signal-to-noise ratio is low, the output volume will be automatically reduced by 16dB. To avoid the above situation, just turn off the soft mute function according to the register position provided by the datasheet.

Q15. Why is the measured isolation always 0 when testing the stereo isolation?

In the above situation, please check and confirm as follows:

1) Confirm that the internal register is not in the internal forced mono demodulation mode;

2) Confirm that the input signal strength is large enough to make the internal demodulator in the stereo demodulation state.

Q16. Why is it found that noise can be received at the frequency of 96MHz in many applications, and how to solve it? In many systems, 12MHz or 24MHz is often used as the system clock, so that their harmonics 96M Just falls within the FM band. To minimize the effect of this harmonic, optimize as follows:

1) Optimize the PCB layout design.

2) If you use the headphone cable as an FM antenna, please add isolation beads;

3) In the system application, when using the FM function, minimize the work of other modules;

4) Minimize the drive capability of the crystal oscillator circuit;

5) When possible, reduce the main frequency of the system clock, so that the harmonic interference is greatly reduced.

Q17. When auto-searching in FM mode, the number received in the low frequency band is less, and the effect is poor; in the high frequency band, it is better, how to solve it?

This phenomenon is because the resonant frequency point of the antenna resonant circuit shifts to the high frequency band, which can be improved by appropriately increasing the resonant capacitor. For the parameter selection of the resonant circuit, please refer to 2.2 "Design of the resonant circuit at the input end of the FM antenna".

Q20. The separation can reach 31dB at 75% modulation, why is the separation only 8dB at 100% modulation? Reg17 (bit7~13) threshold setting is too high, the default setting is 0x0C00, changing it to 0x0800 can solve the problem.

In addition: Reg4 (bit6~8) cannot be set too high, 0x60C0 is recommended.

Q21. Why is BK1088 still separated in forced mono mode, and how to solve it?

When Reg2 (bit12) is set to 1 (forced mono), although the ST flag is 0, there will still be a few dB of separation. At this time, the following changes need to be made:

Reg17 (bit7~13) = 0x7FH, all 1 Reg4 (bit6~8) = 0X7H, all 1

Q22. Under what circumstances is it easier to separate the left and right channels?

The smaller the setting of Reg17 (bit7~13), the easier it is to achieve separation. Conversely, when forcing mono, it is best to set it

to 1.

Q23. Why is the BK1088 ST flag asynchronous with separation?

The BK1088 is different from the BK1080. The BK1080 analyzes the audio signal in stereo after the system detects the stereo station in the current station, and when the signal strength reaches the acceptable definition in the stereo solution. At this time, the S T flag is 1. Note: If the stereo signal is too weak, the sound effect is not good if it is resolved as stereo, and it is generally analyzed as mono. The ST flag of BK1088 indicates whether the current radio station is a stereo station. As long as it is a stereo station, ST is 1. When the signal reaches the resolution that can be solved by stereo and is acceptable, the separation will gradually become stronger. So the ST flag is not synchronized with the stereo split.

Q24. How strong is the stereo signal of the BK1088 to start to separate, and how strong to start to completely separate?

Depending on the register settings, there will be a difference in the separation threshold for different board sensitivities. Take the following settings as an example:

Reg17 (bit7~13) = 0x 0800 The left and right channels should be gradually separated at 16~18dBuEMF. The best separation of the chip is 31dB at around 30dBuEMF.

Q25. Why the left and right channels of BK1088 can only gradually separate as the signal becomes stronger?

1) The design of BK1088 itself is to gradually separate according to the increase of signal strength. This design can make the audio signal we hear clearer when the signal is not very strong. This is also the mainstream design method for AM/FM integrated chips on the market.

2) When the signal is not strong enough and the signal is not completely separated, if you want to achieve a sudden change, you can use forced stereo, that is, set the forced stereo bit Reg2 (bit11) to 1, and the left and right channel separation can immediately reach a maximum separation of about 31dB.

3) It is not difficult to realize the mutation technology of the left and right channels. Our previous single FM chip BK1080 is separated by the mutation separation of the left and right channels.

Q26. The frequency response of high frequency is very poor when tested with a signal generator. The amplitude of 10KHz is 6dB lower than that of 1KHz. Why? FM radio transmission has pre-emphasis processing, so FM reception will do de-emphasis.

The signal sent by the signal generator has no pre-emphasis, so it is necessary to turn off the de-emphasis when receiving to test the real frequency response. That is, Reg7 (bit10) is set to 1 (bypass de-emphasis) during the test.

In non-test mode, de-emphasis needs to be turned on, that is, Reg7 (bit10) is set to 0.

Q27. Why the signal-to-noise ratio of BK1088 is only about 39dB when tested with a millivoltmeter. And the specification indicates that the signal-to-noise ratio can reach more than 50dB?

The BK1088 audio output part contains a harmonic component of about 1Mhz, and the normal audio signal is generally within 15Khz, and the signal of about 1Mhz cannot be released through the power amplifier and the speaker. Therefore, there is no sound on the audio quality, and the separation can be tested with an audio analyzer to reach more than 50dB. However, the analog millivolt meter cannot filter out this part of harmonic components. Therefore, the signal-to-noise ratio tested will be relatively poor. If the customer must see better data on the millivolt meter. A large magnetic bead can be connected to the output of the left and right channels to consume the high frequency components of the audio.

Q28. Why does the automatic channel search often cause frequency deviation, and the AFC data read out are all 0?

AFC: Reg8 (bit7~15) is an important parameter for judging frequency offset. In BK1080, this parameter is called Freqd. The AFC of BK1088 has an enable bit AFCEN which is Reg8 (bit15). Set to 1 to enable.

Defaults to 1 at initialization and is enabled. If the program is running, turn off this bit. The AFC reading will always be 0, resulting in frequent frequency deviations.

Q29. Why does the program suddenly stop responding when it is running? No sound output either?

Reg2 (bit6) of BK1088 is the chip power down enable bit, if this bit is accidentally set to 1, it will make the chip

Go to sleep, check the state of this bit. Also check if the chip mute bit is set.

Q30. The chip stopped at 87.8MHz after the automatic channel search, and ran to 89.8MHz within a few seconds. And obviously

The 87.8MHz is still unchanged, and the clock is not biased. Why?

Please check whether the SEEK bit Reg2 (bit8) is set to 1, if the SEEK bit is set to 1, and the SEEK bit is not cleared during the automatic channel search process, the internal hardware search of the chip will continue to work until a real channel is found , and then stop at the first station found and play again. This is where the problem arises.

Q31. How to choose the BK1086/88 ferrite antenna? How to choose the BK1086/88 antenna? What is the BK1086/88 antenna specification?

The selection of magnetic rod antenna mainly follows the following principles:

1) In principle, the longer and thicker the magnetic rod, the better the effect. But it also depends on the cost, product shape and so on.

2) High Q cores are better than normal cores.

3) In principle, the more coil cores on the magnetic rod, the better, seven cores are better than five cores

4) The number of turns of the coil is not as many as possible. You can wind a certain number of turns (such as 150 turns) first, and then increase or decrease the number of coils according to the effect. to achieve the best results.

In short, the debugging of the magnetic rod antenna requires more professional equipment and experienced technicians. If your company does not have the relevant It is best to bring the relevant equipment and experience to a professional antenna manufacturer for custom-made boards and casings.

Q32. Can the BK1086/88 achieve stereo separation at the specified receiving sensitivity, and if the sensitivity is lower than this sensitivity, it will be resolved in mono?

Can BK1088 do the following functions:

When the stereo signal is lower than 20dBuEMF, it will be decoded as mono, and the LCD will not display the ST bit. When the stereo signal is higher than 20dBuEMF, press stereo decoding, and the LCD will display the ST bit. And the separation will be mutated.

The ST flag needs to change according to the signal change, and it changes once in 2S.

Can the above be achieved?

It can be realized, the process is as follows:

1) Firstly, use RSSI, SNR to lock the 20dBuEMF threshold. The threshold value we measured is roughly as follows, and there will be slight differences depending on the board:

RSSI = 0x1AH SNR = 0x0AH

This is the data obtained in the shielded room. The SNR value of the actual space environment will not be so high, so the SNR can be set to half of 05H. RSSI threshold is more accurate.

2) Read the ST flag every two seconds, and set the relevant registers by default as follows: (1) Force the stereo bit Reg2 (bit11) to be set to 0 (Normal operation state)

(2) Force mono bit Reg2 (bit12) to 0 (Normal operation state)

(3) Reg17 (bit7~13) restore to default 0x10H

(4) Reg4 (bit6~8) restore to default 0x3H

After changing these settings, it needs to delay about 20ms before reading ST, because the system may be in the state of forced mono or forced stereo before changing these registers.

3) Read the ST flag, if the current station is a mono station, then set the system to forced mono. If the current station is a stereo station, but read RSSI, SNR found that the signal is not enough 20dBuEMF, also press the forced mono

Road processing.

The forced mono settings are as follows:

(1) Force stereo bit Reg2 (bit11) to be set to 0 (Normal operation state) (2) Force mono bit Reg2 (bit12) to be set to 1 (Force mono. state) (3) Reg17 (bit7~13) is set to 0x7FH, That is, all 1s.

(4) Reg4 (bit6~8) is set to 0x7H, that is, all 1s.

4) When reading ST is 1, and by reading RSSI, when the SNR threshold finds that the sensitivity reaches above 20dBuEMF, press the forced stereo solution, so as to achieve separation mutation, the settings are as follows:

(1) Set the forced stereo bit Reg2 (bit11) to 1 (Force stereo. Status)

(2) Force mono bit Reg2 (bit12) to 0 (Normal operation state)

(3) Reg17 (bit7~13) restore to default 0x10H.

(4) Reg4 (bit6~8) restores to the default 0x03H.

Execute steps 2) ~ 4) every 2 seconds to achieve the desired function.

Q34. Important note: BK1086/88 automatic channel search parameter AFC?

AFC: Automatic channel search frequency offset judgment parameter, which mainly plays the role of removing adjacent frequency points of real stations.

For example: 89.8MHz is a very strong station in Shenzhen. In addition to the very clear radio sound at the frequency of 89.8MHz, our radio can also hear better sound effects at places with strong signals at 89.7MHz and 89.9MHz.

At this time, if we use simple RSSI (real-time signal strength indication) and SNR (signal-to-noise ratio) to judge whether 89.7MHz and 89.9MHz are real, it is definitely useless, and we can only rely on AFC.

The general principle of AFC: When the radio antenna input receives an 89.8MHz radio signal, the chip will compare the received radio carrier signal with the local oscillator signal generated inside the chip. If the local oscillator signal is also 89.8MHz, AFC The resulting value will be extremely small or extremely large, and the larger or smaller value depends on whether the current radio carrier signal is higher than the local oscillator signal or lower than the local oscillator signal.

When the received radio signal is 89.8MHz, and the set frequency of the radio is 89.9MHz, that is, when the internal oscillator signal of the chip is 89.9MHz, the AFC value obtained by comparison will be relatively middle, not very large nor will be very small.

**BK108X Arduino Library**

The BK108X Arduino Library is designed to provide easy-to-use functions for controlling the BK1088E FM Radio Receiver chip using various Arduino boards and microcontrollers. It offers compatibility with over 26 different boards, including ESP32, Raspberry PI Pico, ATMega 328, Arduino DUE, Aerduino Mega, and STM32.

**Features**

* Control and configure the BK1088E FM Radio Receiver chip.
* Easy integration with Arduino IDE and Arduino-compatible boards.
* Support for various FM radio functionalities, including frequency tuning, volume control, and signal quality measurements.
* Enhanced FM/RDS functions, including error information retrieval and system setup.
* Compatibility with different reference clock frequencies (32.768kHz by default).

**Getting Started**

To begin using the BK108X Library, follow these steps:

1. Download the latest release from the [GitHub repository](https://github.com/pu2clr/BK108X/releases).
2. Extract the library files into your Arduino libraries directory.
3. Open the Arduino IDE and navigate to **Sketch > Include Library > BK108X** to include the library in your sketch.
4. Connect the BK1088E FM Radio Receiver chip to your Arduino board according to the provided pin connections in the library documentation.
5. Use the library functions to control and configure the BK1088E chip in your project.

**Documentation**

Detailed documentation on the library's functions, examples, and usage can be found in the [Wiki](https://github.com/pu2clr/BK108X/wiki) section of the GitHub repository. The documentation covers the available functions, their parameters, and usage examples.

**Contributing**

Contributions to the BK108X Library are welcome. If you find any issues, have suggestions for improvements, or want to add new features, please open an issue in the [issue tracker](https://github.com/pu2clr/BK108X/issues) or submit a pull request.

**License**

The BK108X Arduino Library is released under the [MIT License](https://github.com/pu2clr/BK108X/blob/main/LICENSE). Please review the license file for more details.

**Acknowledgements**

This library is built upon the work and contributions of various developers and contributors. Their efforts are greatly appreciated, and their names can be found in the [Acknowledgements](https://github.com/pu2clr/BK108X#acknowledgements) section of the repository.

**Support**

If you encounter any issues or need assistance while using the BK108X Library, you can reach out for support by creating an issue in the [GitHub repository](https://github.com/pu2clr/BK108X/issues).

We hope you find the BK108X Arduino Library helpful for your FM radio receiver projects. Happy hacking!