HT6xL25x0A Low Power Application Guide

D/N: AN0673EN

Introduction

Low power products can be seen everywhere in present day life, such as for NFC, RF energy harvesting, disposable batteries, solar panel products, etc. It can be seen that low power applications have occupied a large share of embedded systems. Holtek Low Power A/D Flash series of HT66L25x0A and HT67L25x0A MCUs includes both LIRC and LXT oscillators. The standby current for these devices is lower than 1μA, which can meet the demands of power-saving applications and thus extend battery life. The MCU's integrated MIRC oscillator can switch between high frequency and low frequency operation. It can also be used as the clock source for the A/D function to reduce the A/D conversion power consumption. These MCUs include the following advantages:

- The sleep current can be as low as 80nA.
- The watchdog timer current is 500nA.
- The operating current can be as low as 92.5μA/MHz (4MHz).
- The operating voltage can be as low as 1.8V (4MHz and below).

These low current characteristics are suitable for portable applications powered by disposable batteries. This application guide aims to assist system designers to obtain a brief understanding of these devices from both software and hardware aspects. Users can learn how to use these MCU devices and how to configure the low power modes to implement optimal processing in power sensitive applications.

Power Consumption Basic Knowledge

Power Consumption Sources

MCU power consumption can be divided into two parts: dynamic power consumption and static power consumption. Dynamic power consumption, usually referred to as the logic power, is the power that is consumed by the MCU when it is operating and executing instructions. This can be further divided into two parts, namely the switching power consumption and the short circuit power consumption. The static power consumption is the power consumed when the MCU is powered but no instructions are being executed.

Dynamic Power Consumption

The dynamic power consumption is mainly divided into two parts: switching power consumption and short circuit power consumption.

1. Switching power consumption: this can be understood as the power consumed during transistor inversion. This is the power consumed when the load capacitance is charged and discharged during the circuit switching process. When the input is at a low level, the PMOS transistor will be on while the NMOS transistor will be off. Here V_{DD} will charge the load capacitance, as shown in Figure 1. When the input is at a high level, the NMOS transistor will be on while the PMOS transistor will be off, and therefore the load capacitance will discharge.

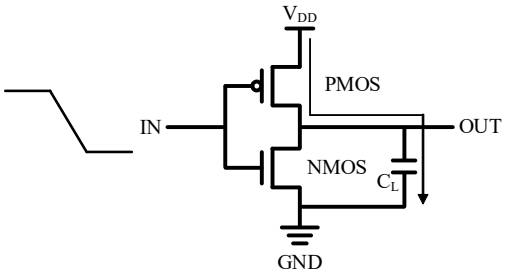

Figure 1. Switching Power Consumption Path

The switching power consumption calculation formula is shown below:

$$P_{switch} = \alpha \times V^2 \times f \times C$$

Where V is the system voltage, f is the switching frequency, C is the load capacitance value and α is the activity factor, which is the probability of a circuit node switching from 0 to 1. Therefore, reducing the switching power consumption can be implemented from the perspectives of the voltage, load capacitance and operating clock frequency. In terms of the voltage, the switching power consumption has a squared relationship with the voltage. Therefore in applications, reducing the voltage can significantly reduce the power consumption. In terms of the operating frequency, reducing the frequency also reduces the dynamic current. However considering that the execution speed is also a factor affecting the power consumption, it is necessary to consider lowering the power consumption by reducing the operating frequency based on the actual application conditions. In terms of the load capacitance, the load capacitance comes from the wirings and transistors in the circuit. Application designers can reduce inter-interface

capacitances to reduce the switching power consumption by optimising the component layout and wiring as well as using multi-chip packages.

2. Short circuit power consumption: this is also known as the internal power consumption. Because the input signal inversion cannot be completed instantaneously, there is a very short transition period during which both the NMOS and PMOS transistors are simultaneously on when the input signal is switching from low to high or from high to low. This creates a path between the power supply V_{DD} and ground GND, resulting in a short circuit current, as shown in Figure 2 below.

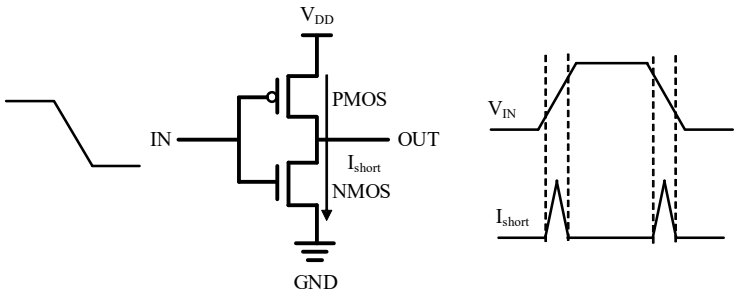

Figure 2. Short Circuit Power Consumption Path

The short circuit power consumption calculation formula is shown below:

$$P_{short} = V \times f \times Q$$

Where V is the system voltage, f is the switching frequency and Q is the charge quantity that flows from the power supply V_{DD} to ground GND during a single signal transition. Here the short circuit power consumption is related to the system voltage and frequency. Therefore, under the condition of meeting the application requirements, the lower voltage, the less power consumption. While a lower frequency can reduce the system power consumption, it reduces the system performance and extends execution time for the same tasks. This means that the power consumption may actually not be reduced. Therefore, designers need to comprehensively consider the relationship between performance and power consumption to find the best optimum point.

Static Power Consumption

The static power consumption refers to the power consumed by partial or all of the chip circuits which are powered but not in operation. The main source of this power consumption comes from subthreshold leakage currents which flow through off-state transistors, the leakage current which flows through the gate dielectric and the PN junction leakage current in the source diffusion region. In applications, the static power consumption is mainly generated when the system is in the SLEEP Mode, while for battery-powered applications, because they are operating in the IDLE mode most of the time, the static power consumption is a main factor to consider.

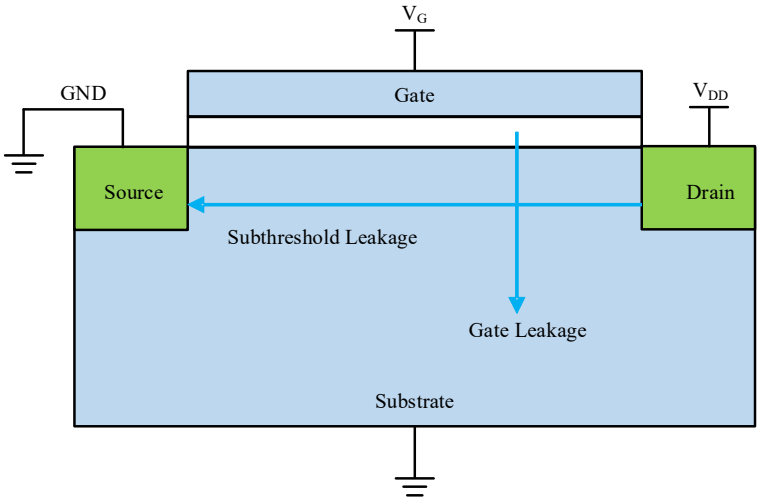

Figure 3. NMOS Transistor Leakage Current

The subthreshold leakage current is related to various factors. Leakage currents can be reduced by increasing the source voltage and adjusting V_{DD}. It is also influenced by temperature and voltage. Additionally, by connecting two or more transistors in series, the transistor leakage current can be significantly reduced due to the stacking effect. The gate leakage current is generated when a voltage is input to the gate and carriers tunnel through the thin gate dielectric. The gate leakage current can be reduced by stacking transistors and placing the off-state transistors close to the power supply line. The PN junction leakage current is generated when the source or drain diffusion region is at a different potential from the substrate. The PN junction leakage current is small compared to other kinds of leakage current. For designers, the best approach in reducing the leakage current is to reduce the voltage and turn off circuits that are not needed.

Based on the aforementioned description of the static and dynamic power consumption, to reduce the system power consumption in practical applications, the following factors can be considered:

- System operating voltage
- System clock
- PCB layout and routing
- Enabled peripheral circuits
- MCU operating modes

Power Consumption Measurement

The system total power consumption mainly includes the average power consumption and maximum power consumption. The average power consumption is equal to the ratio of the total energy consumed by the system for both dynamic and static power consumption modes to the average system cycle time.

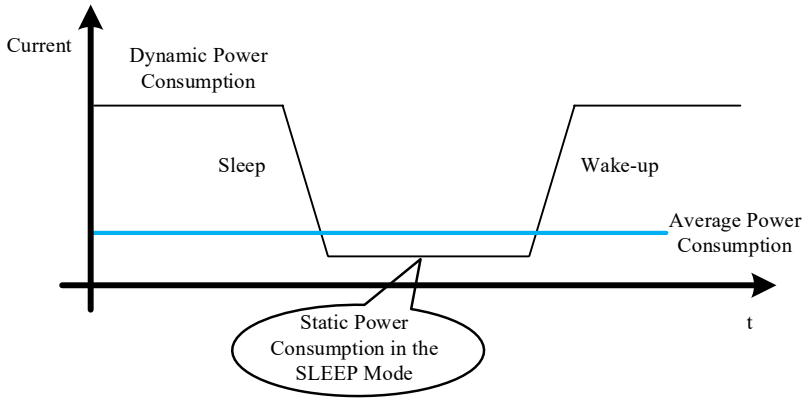

Figure 4. Average System Power Consumption Calculation

$$\text{Average current} = \frac{I(activity)\times t(activity) + I(sleep)\times t(sleep)}{t(activity) + t(sleep)}$$

To measure the dynamic power consumption, a current sensing resistance can be connected in series with the power supply line. The power consumption is calculated by observing the voltage drop across the resistance using an oscilloscope. The current sensing resistance should be of a moderate value. If the resistance value is too large, it will cause the system supply voltage to drop by too large an amount. If the resistance value is too small, it will be difficult for the oscilloscope to observe the voltage drop. Resistances ranging from 10Ω to 100Ω are generally suitable for dynamic power consumption measurements.

The current due to static power consumption is very small, typically in the μA region or even smaller. If using the resistance sampling method to measure, users should use an oscilloscope to observe the voltage drop across the resistance. Taking a current of 1~10μA as an example, users needs to use a sampling resistance of at least 5kΩ to measure the voltage drop. If the system is not in a low power state, a large resistance will affect normal system operation. To accurately measure the current in the low power mode, a high precision ammeter is needed. If a high precision ammeter is not used, the power consumption of the low power system can be measured using the capacitance constant current discharge formula. As shown in Figure 5, connect the capacitance to the system and use a switch to disconnect the capacitance from the power supply and the system.

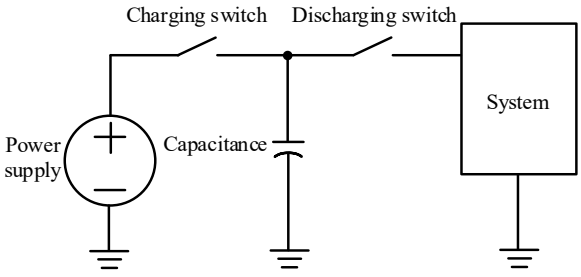

Figure 5. Low Power Consumption Measurement - Capacitance Method

Steps for measuring the power consumption using the capacitance method:

1. Turn on two switches simultaneously to allow the power supply to charge the capacitance until the capacitance voltage reaches V_{DD}. A voltmeter can be used to measure the voltage.

2. Remove the voltmeter first, turn off the power supply and allow the capacitance to power the system for a period of time and then disconnect the capacitance from the system. Ensure that the capacitance voltage is within the normal system operating voltage range before disconnecting.
3. After disconnecting the system, immediately measure the capacitance voltage using a voltmeter.
4. Calculate the capacitance constant current discharge current value based on the voltage drop difference and discharge time using the following formula.

$$I = C \frac{\Delta V}{t}$$

In the above formula, C represents the capacitance value in farad (F) units. ΔV represents the voltage drop across the capacitance in volt (V) units and t represents the capacitance powered time in second(s) units. I represents the capacitance discharge current in ampere (A) units. Assuming that the capacitance used is 100μF, the system is powered by the capacitance runs for 10 seconds, the generated voltage drop is 100mV, users can then calculate that the system average current during 10 seconds is 1μA using the above formula.

It should be noted that when using this method, capacitances with low leakage current should be used. If capacitances with high leakage current are used, the results will have significant errors. If it is unavoidable to use capacitances with large leakage currents, the experiment can be repeated for a longer period of time without a load on the capacitance to determine the leakage current value. When calculating, the leakage current should be included in the above formula.

HT6xL25x0A Low Power Features

For low power applications, Holtek has released the HT6xL25x0A series of MCUs. These MCUs provide flexible clock configurations and operating mode switching functions, which can assist designers to obtain the optimum point between power consumption and performance.

Clock Configuration

Taking the HT67L2550A device as an example, five oscillator frequency sources are provided, LIRC, MIRC, HIRC, LXT and HXT oscillators. Users can select one of them as the system frequency source, f_{SYS}, using the relevant register. The LIRC/LXT oscillators provides a 32768Hz frequency for time counting and because of its low oscillating current, it can be used for standby counting and wake-up functions. The MCU can enter the SLEEP or IDLE power saving modes using the program configuration. The HIRC/HXT oscillators provide higher frequencies which are used for applications such as rapid response and communication clock source applications, etc. The MIRC oscillator provides frequencies of 64/128/256/512kHz, which are located between the low frequency and high frequency. The MIRC oscillator is used for low power operation or to provide an ADC clock source to implement power saving.

Type	Name	Frequency	Pin
External High Speed Crystal	HXT	1MHz~16MHz	OSC1/OSC2
Internal High Speed RC	HIRC	2/4/8MHz	—
Internal Middle Speed RC	MIRC	64/128/256/512kHz	—
External Low Speed Crystal	LXT	32.768kHz	XT1/XT2
Internal Low Speed RC	LIRC	32.768kHz	—

Table 1. Oscillator Types

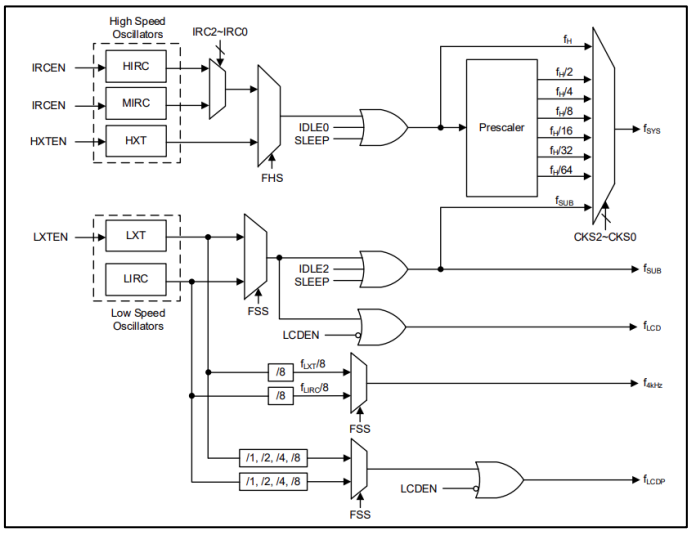

Figure 6. Clock Frequency Block Diagram

The low frequency clock is sourced from either the LIRC oscillator or the LXT oscillator, which is selected using the FSS bit. The LIRC/LXT oscillators provides several frequency paths, which are f_{SYS}, f_{SUB}, f_{LCD}, f_{4kHz} and f_{LCDP}. The f_{4kHz} frequency can be used as a clock source for the WDT and Time Base functions, which can minimise power consumption during counting. When the operating voltage is 3V, the MCU standby current can be lower than 1μA while the LIRC oscillator typical standby current can reach 0.5μA.

Operating Modes

The MCU has six operating modes with different operating modes being selected based on the performance and power consumption requirements, as shown below:

- FAST Mode: the CPU operates normally and the fast speed clock is enabled normally
- SLOW Mode: the CPU operates normally and the slow speed clock is enabled normally
- SLEEP Mode: both the CPU and f_{SUB} are stopped. If the WDT or time base is enabled, f_{4kHz} will continue operating
- IDLE0 Mode: the CPU is stopped and the slow speed oscillator is enabled to drive some of the peripheral functions
- IDLE1 Mode: the CPU is stopped and the high and low speed oscillators are enabled
- IDLE2 Mode: the CPU is stopped and the high speed oscillator is enabled

Among these modes, the low speed and fast speed modes are both modes where the CPU operates normally. The remaining four modes are used to conserve power when the MCU CPU is turned off. In the normal operating modes, all clocks are operating and the CPU is in an operating state. The power consumption in these operating modes positively correlates with the operating frequency of the system clock. If the application does not require a high frequency operation condition, the system clock frequency can be reduced to lower the power consumption. Additionally, different IDLE Modes and the SLEEP Mode can be selected based on actual low power application requirements.

The clock speed is the most important factor when considering dynamic power consumption. It offers significant flexibility in reducing the dynamic current consumption. For instance, a low speed clock can be applied to code sections that are not time-sensitive. The clock can be switched to a high speed clock source when the MCU is processing complex code or if the codes are time-sensitive. Implementing these recommendations can greatly benefit low power applications.

The SLEEP Mode is the most commonly used and flexible mode. In this mode, the wake-up time is very short and it requires little overhead to enter and exit the SLEEP Mode. Therefore, this is the best low power mode for applications that require a short sleep time, a quick wake-up time and rapid processing.

Operating Mode Switching

The system clock can be configured using the CKS2~CKS0 bits in the SCC register to implement mode switching between the SLOW Mode and FAST Mode. The FHIDEN and FSIDEN bits in the SCC register are used to select the IDLE Mode or SLEEP Mode. After the HALT instruction is executed, the MCU will enter the corresponding operating mode.

```
void Idle0_init(void)
{
    _fhiden = 0;
    _fsiden = 1;
    _halt();
}

void Idle1_init(void)
{
    _fhiden = 1;
    _fsiden = 1;
    _halt();
}

void Idle2_init(void)
{
    _fhiden = 1;
    _fsiden = 0;
    _halt();
}

void Sleep_init(void)
{
    _fhiden = 0;
    _fsiden = 0;
    _halt();
}
```

Figure 7. Operating Mode Switching Example Codes

Power Consumption Data in Different Modes

At a voltage of 3V, the MCU current consumption performance in different operating mode configurations is shown in the figures below.

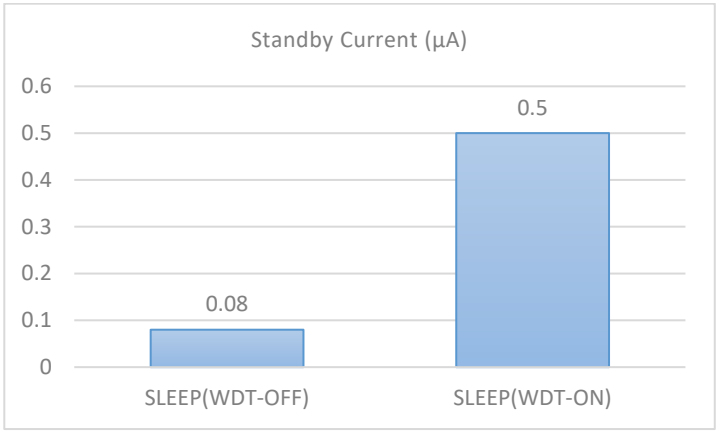

Figure 8. Standby Current at 3V

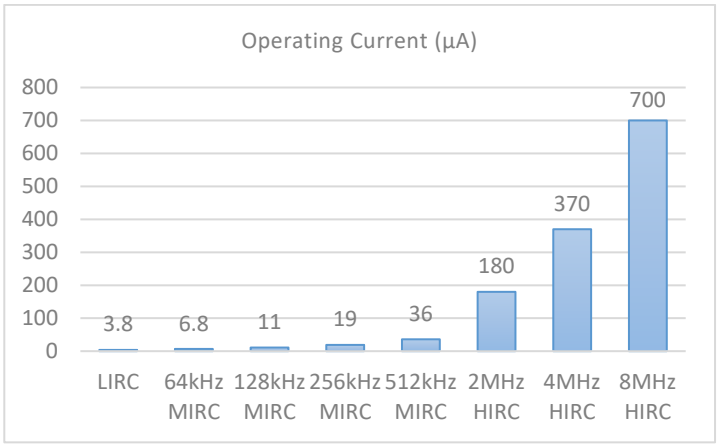

Figure 9. Different Oscillator Operating Current at 3V

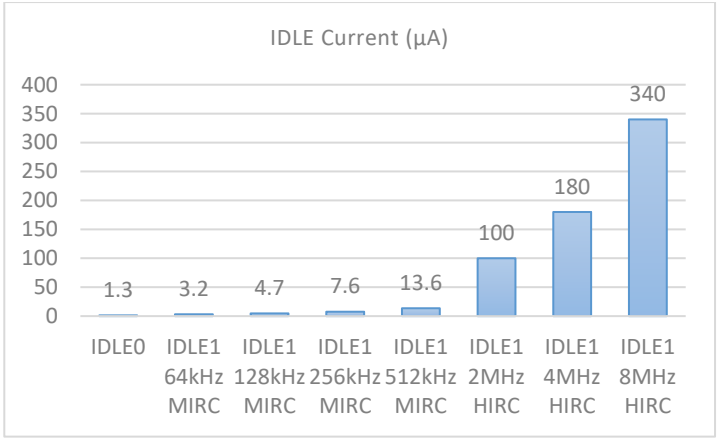

Figure 10. IDLE Mode Current at 3V

Note: refer to the “D.C. Characteristics” section in the MCU datasheet for the standby power consumption and operating current parameters of these modes.

The above tests in various modes are implemented under no-load conditions. The example code are as follows:

```
void main()
{

  Lirc_clok();
  Wdt_off();           //sleep mode  wdt disable
  Sleep_init();

//  Lirc_clok();
//  Wdt_timeout();
//  Wdt_on();            //sleep mode  wdt enable
//  Sleep_init();

//  Lirc_clok();
//  Wdt_Off();           //idle0 wdt disable
//  Idle0_init();

//  Lirc_clok();
//  Wdt_timeout();       //idle0 wdt enable
//  Wdt_on();
//  Idle0_init();

//  hirc_clok();
//  Wdt_Off();           //idle1 wdt disable
//  Idle1_init();

//  hirc_clok();
//  Wdt_timeout();       //idle1 wdt enable
//  Wdt_on();
//  Idle1_init();

//  hirc_clok();
//  Wdt_Off();           //idle2 wdt disable
//  Idle2_init();

//  hirc_clok();
//  Wdt_timeout();
//  Wdt_on();            //idle2 wdt ensable
//  Idle2_init();

  while(1)
  {
    GCC_CLRWDT();
  }

}
```

Figure 11. Test Codes of Different Modes

Wake-up

After the MCU enters the IDLE or SLEEP Mode, it can be woken up via one of following sources:

- An external hardware reset
- A falling edge on Port A
- A system interrupt
- A WDT time-out

The following tests can be used to measure the time required for the system to wake up from the SLEEP Mode using a falling edge on Port A under different system clock source configurations.

Measurement method: when the system wakes up due to a falling edge on Port A, the CPU will execute the next instruction after the HALT instruction. To measure the wake-up time, an observation pin should be setup. Before executing the HALT instruction, this pin should output a low level. After detecting a falling edge on Port A and exiting the HALT mode, the pin will output a high level. Therefore, the time from the falling edge on Port A to the rising edge of the observation pin is the wake-up time.

```
_pawu3 = 1;
_papu3 = 1;
_pac3 = 1;

 _pcc0 = 0;

while(1)
{

    _pc0 = 0;
    _fsiden = 0;
    _fhiden = 0;
    _halt();

    _pc0 = 1;

     delay(10);
}
```

Figure 12. Test Codes

Test results:

Oscillator	Frequency	Wake-up Time
Internal Low Speed RC Oscillator	32.768kHz	332μs
Internal Middle Speed RC Oscillator	512kHz	88.2μs
Internal High Speed RC Oscillator	8MHz	3.54μs

Table 2. Wake-up Time of Port A Falling Edge

If the system clock is selected from a low speed oscillator, the MCU will have a low operating frequency and low power consumption, but will require a longer wake-up time. Conversely, a high speed clock source offers a shorter wake-up time. Users can configure their optimal solution based on actual application needs.

Power Consumption Optimisation Strategies

From both software and hardware design perspectives, there are several commonly used strategies to optimise the power consumption. For more information about the low power features and application methods of the HT6xL25x0A series, such as the ADC low current mode, low power power-on and reset, refer to other application notes on the Holtek website.

Software Design

Because software errors are not as easy to be found as hardware ones and there is no standard to determine the software low power characteristics, the importance of software in low power applications is often overlooked. Nevertheless, designers still should apply the low power characteristics to the software in applications to reduce power consumption.

Discretionary Peripheral Usage

Careful use of the MCU peripherals can effectively reduce the power consumption. For example, analog-to-digital converters (A/D) are essential analog modules. However, because the power consumption of these analog circuits usually accounts for the majority of the A/D power consumption, using a faster ADC conversion clock and disabling the ADC during sampling can implement lower power consumption.

In addition, the ultra-low-power HT6xL25x0A series MCUs offer two kinds of A/D converter resolution, 10-bit and 12-bit. When the design requirements are met, using the 10-bit resolution will consume less power than the 12-bit resolution. For other peripherals with similar considerations, refer to the corresponding electrical characteristics section in the datasheet and only use the peripheral functions when necessary based on the application needs.

Use Interrupts Instead of Polling

Whether a program uses an interrupt or polling method is not important for some simple applications, however there are significant differences in terms of their low power characteristics. When using the polling method, the CPU has to continuously examine the I/O ports or registers,

making it unable to efficiently execute tasks which results in additional power consumption. However, if using the interrupt method, the CPU does not need to be active and can even enter a power saving mode, to reduce the system power consumption.

CPU Computation Reduction

Reducing the CPU computation can be implemented in many ways. For example, users can precompute some results and store them in the flash memory as well as using lookup tables instead of real-time computations. These actions can effectively minimise the CPU computation and lower the CPU power consumption. For necessary and real-time computation, the computation can be stopped when the desired precision is achieved to avoid excessive computation. Users should use short data types, such as 8-bit character data instead of 16-bit integer data and use integer operations rather than floating point operations, etc.

Unused I/O Setup

After the MCU is powered on, the GPIOs will default to an input floating state. To avoid Schmitt Trigger power consumption caused by external floating GPIOs, unused I/O pins should remain unconnected and be configured to be either in an output or input state. When configured in an output state, the pin will output a fixed state, either high or low level. When configured as an input state, use an external resistor to pull the pin up or down. If the MCU has internal pull-high/pull-low resistors, configure the input pins to be either high or low through programming. This is because, if the pin does not have a fixed voltage level, it may increase the MCU leakage current.

Hardware Design

When designing a low power system, the hardware circuit design has a significant impact on the power consumption. The following points should be noted.

Appropriate Adjustments to Path Impedance

When using the PWM to drive LEDs as an example, using the PWM to drive LEDs with a lower duty cycle can dynamically control brightness and control the LED power consumption to a certain extent. In addition, to meet with the LED brightness requirement, users can increase the size of the current limiting resistor to drive LEDs with a lower current to reduce the LED power consumption.

Leakage Current Reduction

When current flows through a capacitance, a small amount of charge will be lost. This loss is called the “capacitance leakage current”. The size of the leakage current is related to the size and type of the capacitor. Generally speaking, tantalum and electrolytic capacitances have higher leakage currents, while ceramic and film capacitors have lower leakage currents.

The capacitor leakage current calculation formula is shown below:

$$\text{I(nA)} = \frac{V \times C}{IR(M\Omega \times MF)}$$

IR is the capacitor insulation resistance, V is the voltage across the capacitor and C is the capacitor value.

Therefore, in designing a low power system, it is recommended to choose capacitors with low leakage currents.

Peripheral Power Supply Optimisation

When examining the entire system, its power consumption may come from various sources of which the MCU power consumption is just one of them. Optimising the power consumption of peripheral circuits is also an important aspect.

Taking a voltage divider temperature sampling circuit using an NTC as an example, a common circuit is shown in Figure 13. Assuming that the system is powered by a 3V voltage, when the MCU is in the SLEEP Mode, its voltage divider circuit will generate a current of approximately 30μA, which is significant in a low power system. By modifying the voltage divider circuit and using an MCU I/O port to control the voltage divider power supply, as shown in Figure 14, the power consumption can be reduced. When temperature sampling is required, PD1 can be controlled to power the voltage divider circuit. Before the MCU enters the SLEEP Mode, the power supply to the voltage divider circuit should be disconnected first to implement power saving. It should be noted that this method is only suitable for low power components.

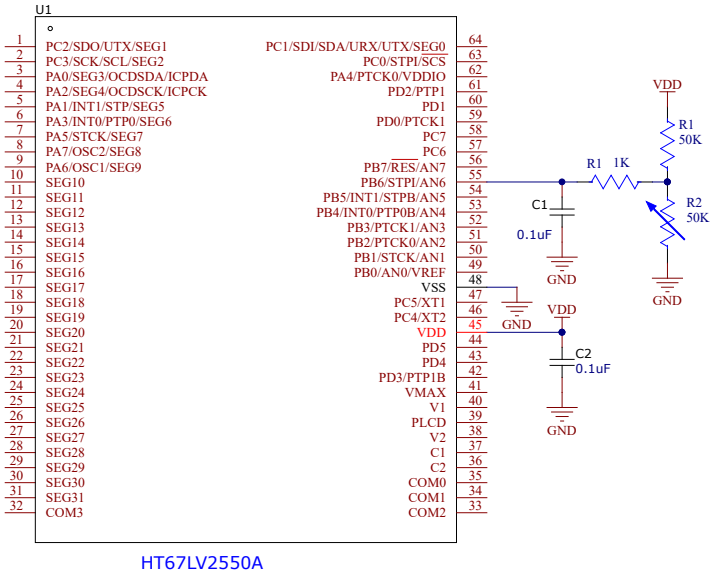

Figure 13. Not Optimised Voltage Divider Circuit

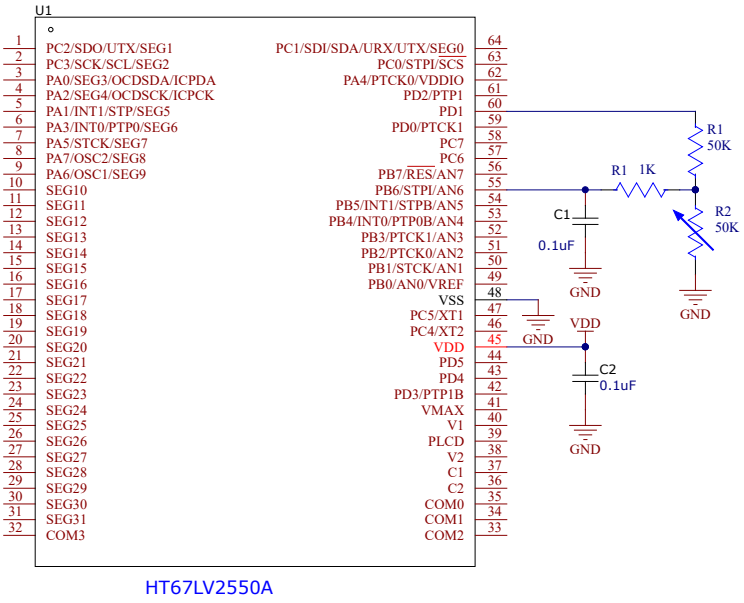

Figure 14. Optimised Voltage Divider Circuit

Conclusion

This application guide has introduced the standby low power function operation, operating current characteristics in different modes and the power consumption measurement characteristics of the HT6xL25x0A series. It also briefly introduces how to implement low power applications from both software and hardware perspectives, providing useful reference for users.

Reference File

Reference Files: HT66L2530A/40A/50A and HT67L2540A/50A Datasheets

For more information, consult the Holtek official website: www.holtek.com.

Revision and Modification Information

Date	Author	Release	Description
2023.07.05	容昭濱	V1.00	First version

Disclaimer

All information, trademarks, logos, graphics, videos, audio clips, links and other items appearing on this website ('Information') are for reference only and is subject to change at any time without prior notice and at the discretion of Holtek Semiconductor Inc. and its relevant companies (hereinafter 'Holtek', 'the company', 'us', 'we' or 'our'). Whilst Holtek endeavors to ensure the accuracy of the Information on this website, no express or implied warranty is given by Holtek to the accuracy of the Information. Holtek will bear no responsibility for any incorrectness or leakage.

Holtek will not be liable for any damages (including but not limited to computer virus, system problems or data loss) whatsoever arising in using or in connection with the use of this website by any party. There may be links in this area, which allow you to visit the websites of other companies. These websites are not controlled by Holtek. Holtek will bear no responsibility and no guarantee to whatsoever Information displayed at such sites. Hyperlinks to other websites are at your own risk.

Limitation of Liability

In no event shall Holtek Limited be liable to any other party for any loss or damage whatsoever or howsoever caused directly or indirectly in connection with your access to or use of this website, the content thereon or any goods, materials or services.

Governing Law

The Disclaimer contained in the website shall be governed by and interpreted in accordance with the laws of the Republic of China. Users will submit to the non-exclusive jurisdiction of the Republic of China courts.

Update of Disclaimer

Holtek reserves the right to update the Disclaimer at any time with or without prior notice, all changes are effective immediately upon posting to the website.