

Inertial Navigation Attitude Solution for Quadrotor Flying Robots

Zhao Li

School of Integrated Circuits
and Electronic Information
Zibo Polytechnic University
Zibo 255300, PR China

Abstract: Quadrotor flying robots, valued for their VTOL capability, agility and compact design, underpin diverse missions such as aerial photography, inspection, delivery and search-and-rescue. Their endurance, however, is fundamentally bounded by the onboard battery, and the mechanical structure of the quadrotor gives it only a poor payload capability, so the flight time cannot be extended simply by enlarging the pack. Two routes are therefore available: raising the capacity and efficiency of the battery, or managing the remaining capacity precisely and in real time. This paper follows the second route and develops a battery monitoring system for a quadrotor flying robot.

Keywords: Battery Monitoring System; Battery Fuel Gauge; Bq2060; State of Charge (SOC); I2C Bus; SMBus; TWI Bus; Atmega2560;

1. BATTERY MONITORING CHIP

For a quadrotor flying robot, the endurance of its own battery is an essential factor that must be taken into account in the design process. Because the mechanical structure of the quadrotor flying robot determines its poor payload capability, two ideas are generally adopted to prolong its air time. The first is to increase the capacity and efficiency of the battery; the second is to manage the remaining capacity of the battery precisely and in real time. Improving the capacity and efficiency of the battery is currently a research hotspot in engineering, but it is not closely related to this subject. Therefore, in this subject the extension of the air time of the quadrotor flying robot is approached by managing the battery capacity precisely and in real time.

1.1 Selection of Battery Charge Monitoring Chip

The quadrotor's battery pack consists of four 3.1 V lithium-ion cells connected in series, resulting in a nominal voltage of 12 V. Therefore, the selected monitoring chip must have a measurement withstand voltage exceeding 12 V. Furthermore, since the battery is composed of four cells in series, and most state-of-charge (SOC) monitoring chips require precise knowledge of each individual cell's current charge level, it is essential to choose a chip capable of detecting more than four cells. In summary, after consulting the product selection guides of companies such as TI and ADI, we decided to use the BQ2060 battery-monitoring IC manufactured by TI [6]. This device features a 36 V withstand voltage and can monitor up to four cells. The BQ2060 supports charging-control functions and intelligent battery-data commands, communicating with the MCU via either the HDQ16 or SMBus protocol. It measures a range of battery-state parameters, including temperature, remaining capacity, estimated runtime, current, and voltage. Additionally, the BQ2060 includes an LED driver that enables a coarse local indication of the current battery-level status using a four-segment LED display, with each segment representing 25% of the total capacity.

1.2 Pin definitions of the Bq2060

The Bq2060 is housed in an SSOP package with 28 pins in total. The pin definitions are shown in Figure 1, and the function of each specific pin is described below in pin order.

HDQ16: serial communication input/output interface; it is an open-drain bidirectional communication port used for communication between the fuel-gauge chip and the main controller. ESCL: clock line of the I2C-interface EEPROM memory, used together with ESDA for communication between the Bq2060 and an external EEPROM. ESDA: data line of the I2C-interface EEPROM memory, used together with ESCL for communication between the Bq2060 and an external EEPROM. RBI: register backup input, which supplies backup power to the Bq2060 through a capacitor or a battery during periods of low operating voltage. REG: power-control output; when the battery is used as the operating power supply of the fuel-gauge chip, this power FET is controlled to regulate the battery voltage into the supply voltage of the Bq2060. Vout: EEPROM power-supply output, which supplies energy to the external EEPROM memory. Vcc: chip supply power. Vss: chip ground. DISP: display-control input; at a high level the segmented-LED display function is enabled, and at a low level the segmented-LED display function is turned off. LED1–LED5: LED display segment outputs; when the local segmented display of battery capacity is used, each output can drive one external LED. DFC: discharge-FET control, which controls the discharge FET of the lithium-battery protection circuit and disconnects the supply circuit when the battery capacity is insufficient, so as to prevent the lithium battery from being damaged by over-discharge. CFC: charge-FET control, which controls the charge FET of the lithium-battery protection circuit and disconnects the charging circuit when the battery is fully charged, so as to prevent the lithium battery from being damaged by over-charge. CVON: cell-voltage-divider control, which controls the switching of the external voltage-monitoring FET in order to detect the voltage of an individual cell. THON: battery-temperature control, which controls the external FET to disconnect the operating circuit when the battery is overheated. TS: input terminal for battery-temperature detection, used to detect the operating temperature of the battery. SRC: current-sense input, used to measure the instantaneous current. SR1–SR2: floating charge-sense registers, used to measure the battery voltage and the magnitude of the discharge current. VCELL1–VCELL4: cell-voltage inputs, used to measure the voltages of the single cells connected in series. SMBD: data line of the SMBus

communication protocol. SMBus: clock line of the SMBus communication protocol.

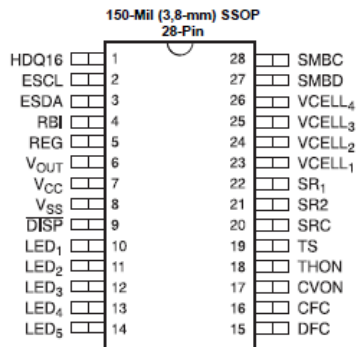


Figure. 1 Pin definitions of the Bq2060

2. I2C BUS, SMBUS BUS AND TWI BUS

The I2C (Inter-Integrated Circuit) bus, created by Philips Semiconductors, is a common two-wire serial interface for short-distance chip communication. Using SDA and SCL lines, it supports multiple devices on one bus with 7-bit or 10-bit addresses. It needs pull-up resistors and works in half-duplex mode, widely used for sensors, EEPROMs and RTC chips. SMBus (System Management Bus) is an I2C derivative optimized for system management. It keeps the two-wire structure but adds stricter timing, electrical rules and timeout protection to prevent bus deadlock. It is mainly used for motherboard monitoring and battery management, and is largely compatible with I2C hardware. TWI (Two-Wire Interface) is functionally equivalent to I2C. Chip vendors like Atmel use this name to avoid trademark issues, with only minor register differences in MCUs. In short, I2C is the base standard; SMBus is its enhanced version, and TWI is another name for I2C.

2.1 Basic introduction of the three communication buses

The I2C bus [2], proposed by NXP, is a two-wire data communication bus (serial data SDA and serial clock SCL). All devices are connected to the bus at the same time, and information is transferred between devices through a master-slave communication mechanism. Each device uses a unique identification address. During the whole communication process, any MCU with an I2C bus controller, any memory with an I2C interface, or any other device with an I2C interface can be defined as either a transmitter or a receiver. The definitions of transmitter and receiver, however, are not fixed: a certain I2C device may be both a transmitter and a receiver. For example, an I2C-interface memory can both receive and send data, so it can be regarded as both a receiver and a transmitter; whereas an I2C-interface A/D can only send the digital signal converted from an analogue signal, so it can only be regarded as a transmitter. Besides the definitions of transmitter and receiver, each device is also defined as a master or a slave. Unlike the transmitter/receiver definitions, the master/slave definitions are fixed. Generally, a device with an I2C bus controller is regarded as a master. During the whole communication process, the master, by coordinating the SDA and SCL lines, issues control signals such as the start and stop of communication to communicate with the slave, and is in the active position during the communication. The data transmission rate on the I2C bus has three modes—standard mode, fast mode and high-speed mode—and the communication rates defined in these three modes are 100

kbit/s, 400 kbit/s and 3.4 Mbit/s respectively. Normally, only the standard mode and the fast mode are needed to satisfy the requirements. The number of devices connected to the I2C bus is limited by a maximum bus capacitance of 400 pF.

The SMBus [3] is also a two-wire interface, proposed by Intel, through which devices can communicate with each other. Its design idea is based on the operating principle of I2C. The SMBus is generally used as a control bus in power-management-related tasks. The differences between the SMBus and the I2C bus lie mainly in the level standard, the load-driving capability, the communication protocol and the maximum communication rate. As for the level standard, in the standard mode of the I2C bus a voltage below 0.3 V_{CC} is defined as a logic 0 and a voltage above 0.7 V_{CC} as a logic 1; the SMBus, in contrast, specifies a logic 0 below a fixed level of 0.8 V and a logic 1 above 2.1 V. As for the load-driving capability, the I2C bus specifies that the leakage current of every device connected to the bus is at most 10 μA, whereas that of the SMBus is 5 μA; in addition, the I2C bus specifies that the maximum number of devices connected to the bus is determined by the capacitance of each device to ground and does not exceed 400 pF, whereas the SMBus has no such provision. As for the communication protocol, the difference is only slight: the SMBus requires the clock frequency on the SCL line to be at least 10 kHz except in the bus-idle state, whereas the I2C bus has no such requirement. As for the maximum communication rate, the I2C bus has three modes and its maximum rate can reach 3.4 Mbit/s, whereas the maximum rate of the SMBus is 100 kbit/s, which is equivalent to communication in the standard mode of the I2C bus.

The TWI [4] bus is a two-wire interface proposed by ATMEL. It is also built on the basis of the I2C bus and is compatible with I2C communication. Its compatibility with I2C is even more thorough: the level standard, the load-driving capability and the communication rate are all completely consistent with those of I2C. What slightly differs from the I2C bus is that the TWI bus introduces a status register to indicate whether the bus is currently idle, which can speed up the communication rate and at the same time makes the operation of the TWI bus more flexible than that of the I2C bus.

In this subject, the Atmega2560 single-chip microcomputer of the Atmel AVR series is chosen as the main control platform. Its two-wire external interface follows the TWI bus standard, whereas the external interface of the Bq2060 chip follows the SMBus standard. Both were developed on the basis of the I2C bus and are compatible with the timing standard of the I2C bus in standard mode. Therefore, when programming the software, the I2C timing standard is uniformly adopted for the communication between them.

2.2 Basic rules of I2C bus data transmission

SDA and SCL are both bidirectional lines, and both are connected to a positive supply voltage through a current source or a pull-up resistor, as shown in Figure 2. When the bus is idle, both lines are at a high level. The output stage of the devices connected to the bus is required to be an open-drain or open-collector circuit, so that the transition between high and low levels can be realized in cooperation with the pull-up resistor.

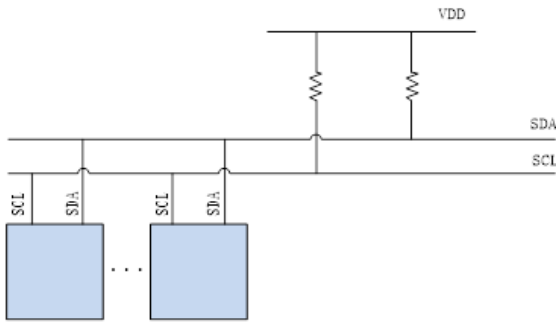


Figure. 2 Devices connected to the I2C bus

When data are transmitted on the I2C bus, the condition that guarantees the validity of the data is that the data on the SDA line are operated according to the level of the clock on the SCL line. When SCL is at a high level, the data on the SDA line are valid; only when SCL is at a low level may the data on the SDA line change, as shown in Figure 3.

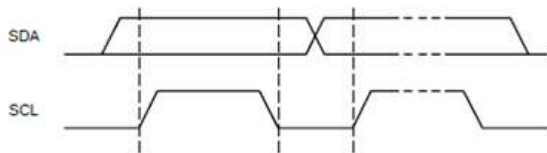


Figure 3 The only valid condition for the validity of I2C bus data transmission

As shown in Figure 4, Start and Stop are the start and termination marks of I2C bus data communication. When the SCL line is high, the Start and Stop states are represented by changing the level signal of the SDA line (a falling edge is Start and a rising edge is Stop). When the master issues a Start state on the bus, transmission begins; when the master issues a Stop state, transmission stops. Between the Start and Stop states, the bus is considered busy, and no other master can take over control of the bus. A special situation occurs when a new Start state is issued between the Start and Stop states. This is called a Repeated Start state, and it is used when the master wishes not to give up control of the bus and to start the next new transmission. After the Repeated Start, the bus remains busy until the next Stop. The Start and the Repeated Start can be regarded as the same starting action; therefore, unless otherwise stated, the Start state and the Repeated Start state are generally not distinguished.

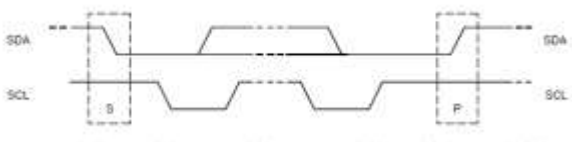


Figure 4 Determination of the Start and Stop states of the I2C bus.

2.3 Flow of I2C bus data communication

The process of completing a data communication on the I2C bus consists of the following parts: the master sends a Start signal, the master sends the address of the slave to be accessed, the slave acknowledges, the master sends the data to be transmitted, the slave acknowledges, and the master sends a Stop. After the Start signal is generated, the bus enters the busy state, and only one clock cycle after the Stop signal is issued does the bus return to the idle state.

There are two ways of transmitting data on the I2C bus: one is that data are sent from the master to the slave, and the other is that the slave responds to the master and returns data to the master. These are the write operation and the read operation commonly referred to.

DA0, DA1, DA2 and DA3 are the inherent address codes of the device; they are fixed inside the chip when the device leaves the factory, and for devices of the same type these four values are identical. A0, A1 and A2 are pin addresses; in the circuit, A0, A1 and A2 are generally connected to ground or to the supply, being 0 when connected to ground and 1 when connected to the supply. The R/W bit is the data direction: when this bit is 1, the slave transmits and the master receives; when it is 0, the master transmits and the slave receives.

According to the provisions of the I2C protocol, the data transmitted on the bus must be transferred in units of bytes, that is, the data unit (byte) sent and received in each master Start/Stop cycle must be 8 bits. Within each master Start/Stop cycle, however, several data units (bytes) may be sent, and each data unit only needs to ensure that the high bit (D7) comes first and the low bit (D0) comes last. After receiving one byte of data each time, the slave must output a low level on the SDA line to generate an acknowledgement signal. For this acknowledgement signal, the slave uses, on the SCL line, the clock signal generated by the master, corresponding to the ninth clock cycle after the Start, and on the SDA line the slave itself has to pull the level low to realize it. This requires the master, after sending one byte of data, to give up control of the SDA line immediately and hand it over to the slave. Sometimes, however, it happens that the slave is exactly busy when the master has finished sending one byte of data, and it cannot accept the data of the new byte. In this case the SCL line is pulled low by the slave, so that the master cannot generate a pulse signal in the next clock cycle and thus enters the waiting state, until the slave has finished other tasks, releases the SCL line, and the master generates a new pulse signal to continue the data transmission. When one byte of data has been transmitted, if the master receives a non-acknowledgement signal (ACK = 1 as defined by I2C) sent from the slave, the master immediately issues a Stop signal to terminate this transmission, or issues a Repeated Start signal to begin a new data transmission. After the master has received the last byte of data sent from the slave, the master sends a non-acknowledgement signal (ACK = 1) to the slave to inform it that the communication has ended and that it need not wait for further information from the master.

3. HARDWARE DESIGN OF THE BATTERY MONITORING SYSTEM BASED ON THE BATTERY MONITORING CHIP

The whole system consists of three parts: the power-supply module, the communication module and the battery fuel-gauge module. Only the schematic design is introduced in this part. No separate PCB was made for the battery monitoring system; instead, it was fabricated on the same PCB as the control system of the quadrotor flying robot.

3.1 Power-supply module

For the chip power-supply part, in order to reduce the complexity of the design, the method of drawing power from the battery through the REG pin of the Bq2060 is not used. Instead, the LM2596 produced by TI and the AM1117-3.3 step-down chip of NXP are used to step down the battery voltage (12 V) to 5 V through the LM2596 and then to the

voltage required for the normal operation of the Bq2060 (3.3 V) through the AM1117-3.3, so as to supply the chip with the electric energy required for its normal operation. The LM2596 [5] module also supplies +5 V to the whole flight control system of the quadrotor flying robot. Its schematic design is shown in Figure 5.

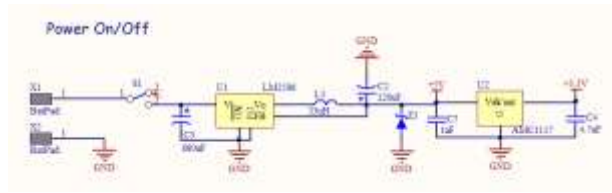


Figure 5 Power-supply module

3.2 Electricity metering module

The electricity metering module consists of the fuel-gauge chip Bq2060, an off-chip EEPROM and local battery-capacity display LEDs. The Bq2060 chip not only contains the battery fuel-gauge function, but also has functions such as battery voltage monitoring and battery temperature monitoring. The off-chip EEPROM is mainly used to store battery-related information. The local battery-capacity display LEDs indicate the current state of battery use locally. Its schematic design is shown in Figure 7.

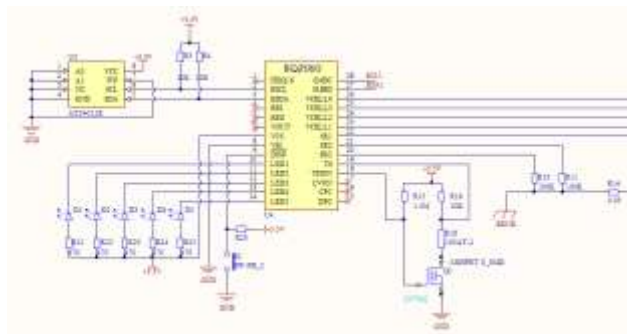


Figure 7 Electricity metering module

4. CONCLUSION

In summary, this paper has studied the hardware design of the battery monitoring system for a quadrotor flying robot. The Bq2060 chip of TI was selected as the main chip of the battery monitoring system, and the hardware schematic design of the whole battery monitoring system was completed and the PCB was fabricated. The I2C bus protocol was analysed, and the similarities and differences among the TWI, SMBus and I2C buses were compared. For the Atmega2560 platform, the driver of the I2C bus communication was given. Accurate and real-time battery-capacity management, obtained in this way, provides a dependable energy basis for the flight control of the quadrotor flying robot and thus contributes directly to the extension of its air time.

5. ACKNOWLEDGMENTS

The author is grateful for the reverent help of technical staff and team support provided by School of Integrated Circuits and Electronic Information, Zibo Polytechnic University.

6. REFERENCES

- [1] Texas Instruments. 2004. Bq2060 Multi-Chemistry SBS 1.1-Compliant Gas Gauge With 5 LED Drivers, Datasheet SLUS035. Texas Instruments Incorporated.
- [2] NXP Semiconductors. 2014. UM10204: I2C-bus Specification and User Manual, Rev. 6. NXP Semiconductors.
- [3] Intel Corporation. 2000. System Management Bus (SMBus) Specification, Version 2.0. Intel Corporation.
- [4] ATmega640/1280/1281/2560/2561 Data Sheet: 8-bit AVR Microcontroller with 64K/128K/256K Bytes In-System Programmable Flash. Atmel Corporation.
- [5] Texas Instruments. 2021. LM2596 SIMPLE SWITCHER Power Converter 150 kHz 3 A Step-Down Voltage Regulator, Datasheet SNVS124. Texas Instruments Incorporated.
- [6] Atmel Corporation. 2010. AVR315: Using the TWI Module as I2C Master, Application Note. Atmel Corporation.
- [7] Atmel Corporation. 2006. AVR311: Using the TWI Module as I2C Slave, Application Note. Atmel Corporation.
- [8] SBS Implementers Forum. 1998. Smart Battery Data Specification, Revision 1.1. SBS Implementers Forum.
- [9] SBS Implementers Forum. 1998. Smart Battery Charger Specification, Revision 1.1. SBS Implementers Forum.
- [10] Texas Instruments. 2022. Battery Management and Fuel-Gauge Selection Guide. Texas Instruments Incorporated..