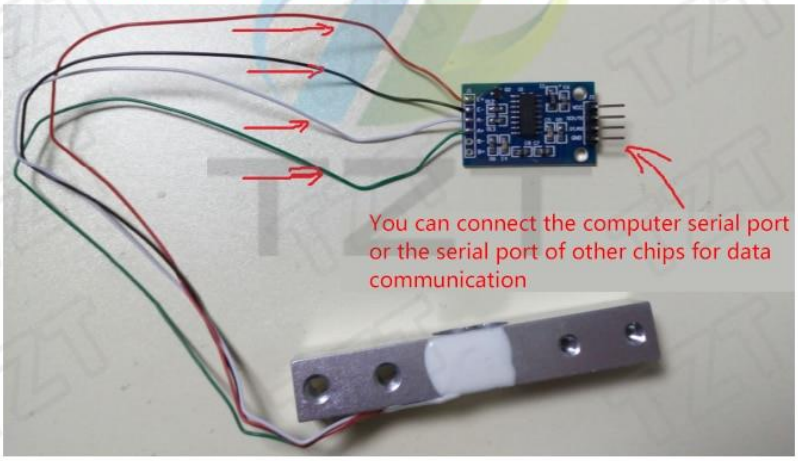
PRODUCT Hx711 Blue Board (AD Serial Port Module)

Main features of electronic scale (hx711) module:

- AB dual channel data acquisition, data transmission with active transmission and passive transmission; Data types are divided into working mode 1 and working mode 2, which can be selected by customers to meet the needs of different users.
- default baud rate: 9600bps check bit: no stop bit: 1 bit total length of protocol: 10 bytes
- volume: length 3.3cm * width 2cm * height 0.3cm.
- the new model adds 47uF black large capacitance tantalum capacitance, which significantly improves the stability.
- built in single chip microcomputer, peeling, correction and other n command control, powerful function. Save bottom development time.
- the voltage stabilizing circuit in hx711 chip can directly supply power to the external pressure sensor.
- simple serial port ttl232 communication, which can be connected to the computer through USB to TTL line for data transmission, which is very convenient.
- synchronous suppression of 50Hz and 60Hz power supply interference.
- the power consumption is very small, and the typical working current is 12mA
- working voltage range: DC 4.7 ~ 5.5V.
- operating temperature range: - 20 ℃ ~ + 85 ℃

Let's take the most commonly used A-channel data acquisition with a gain of 128 as an example （ The module is a dual channel data acquisition, and channel B can also be connected to a pressure sensor. However, the gain of channel B is only 32 times, so the measurement accuracy is not very high. If the accuracy requirement is not very high, AB dual channel can be used at the same time, that is, two pressure sensors can be connected. Channel B is recommended for system parameter detection including battery.)

Most welding colors on the market are red, black, white and green. Of course, some other manufacturers are different. Please see your own sensor lead instructions when welding!



Working mode 1:

Active transmission of channel a; Working mode 1: (factory default format)

Default baud rate: 9600bps check bit: no stop bit: 1 bit

Total protocol length: 10 bytes

Bit1	Start bit	0XAA (fixed)
Bit2	Command format return	0xa1 performs passive transmission (one transmitter and one receiver) on behalf of channel a working mode 1 0xa2 sends actively on behalf of channel a working mode 1 ----------0xb1 performs passive transmission (one transmitting and one receiving) on behalf of channel B working mode 1 0xb2 performs active transmission on behalf of channel B working mode 1
Bit3	Reserve	0X00
Bit4	Reserve	0X00
Bit5	Current channel data low 16 bits	Corresponding to the data returned under bit2 command Format restore ad 24 bit conversion of hx711 【000000H--FFFFFFH】 Data = bit5 * 65536 + bit6 * 256 + bit7
Bit6	High 8 bits of current channel data	
Bit7	Current channel data low 8 bits	
Bit8	Checksum high 8 bits	Check code= (bit2+bit3+....+bit6+bit7)
Bit9	Checksum low 8 bits	
Bit10	End bit	0XFF (fixed)

Note: bit2 factory defaults to A2 command, and actively sends data every 100ms.

In some working environments, if the module does not need to actively send data, you can send 0xa1 to the module, then cancel the active sending command, the module will immediately change to passive sending, send 0xa1 once, and the module will return to collect data once. Similarly, when you want to switch to active sending, you only need to send it once. 0xa2 will complete the switching, and the parameters will be powered off and saved. There is no need to reset next time!

A1 command, the module returns the following format:

Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10
0XAA	0XA1	0X00	0X00	Low data 16 bit	Data high 8 bits	Low data 8 bit	Check bit High 8 bits	Check bit Lower 8 bits	0XFF

Working mode 2:

In order to better understand working mode 2, let's first understand the weighing principle:

Many people think that when there is no measured object on the load cell, the data output is 0. This idea is wrong. In fact, when the measured object is not placed on the load cell, there is still data output, which is related to the internal bridge circuit of the sensor. Moreover, after being processed by hx711ad module, not only this value is not 0, but also this value is very large!

So how do we calculate the weight of the weighing object?

Generally, it is necessary to collect the fur weight when there is no object on the sensor tray.

[step ①] weight_ Maopi = HX711_ Read();// Collect the ad value of fur weight once in advance

After the fur weight is collected, it needs to be stored in the ROM section in the

equipment. Then the next time you weigh

[step ②] weight_ Shiwu = HX711_ Read(); // Collect the ad value of the current weighing, which is equivalent to returning under A1 and B1 commands weight ad value of

[step ③] weight_ jingzhong = Weight_ Shiwu - Weight_ Maopi;// The difference of subtraction is equal to the net weight ad value, and channel B works

In mode 2, there is only the peeling and zeroing command, that is, to obtain the net weight ad value here, and there is no function of the correction coefficient in step 4. Please note that!

[step ④] actual weight = weight_ jingzhong/K; // K is a coefficient. This line is equivalent to the return of channel a working mode 2 command weight value returned, not ad value. It has been converted to actual weight.

In this way, the actual weight of the measured object is obtained.
Just touched the pressure sensor and saw if it was a little dizzy here. Don't worry. Slowly understand the logical sequence. It's still very simple

The above [step ④] is the output data of working mode 2 of channel a in the module. The calculation process ① ~ ③ is directly omitted and the actual weight is directly output

<table>
<tr><td>Bit1</td><td>Start bit</td><td>0XAA (fixed)</td></tr>
<tr><td>Bit2</td><td>Command format return</td><td>0xa3 performs passive transmission (one transmitter and one receiver) on behalf of channel a working mode 2

0xa4 sends actively on behalf of channel a working mode 2

0xb3 performs passive transmission (one transmitting and one receiving) on behalf of channel B working mode 2

0xb4 performs active transmission on behalf of</td></tr>
<tr><td>Bit3</td><td>Reserve</td><td>0X00</td></tr>
<tr><td>Bit4</td><td>Positive and negative flag bit</td><td>0x00: positive 0x01: negative</td></tr>
<tr><td>Bit5</td><td>16 bits lower than actual weight</td><td rowspan="3">Actual weight of channel a = bit5 * 65536 + bit6 * 256 + bit7

The default unit is g. For a large range sensor, if it is accurate to g, the display fluctuation will be large. It is recommended to take 0.1kg or 1kg.

In channel B, the returned data is the ad value after peeling. This channel is not commonly used for weight measurement and is not divided by the coefficient K, so it is the ad value.</td></tr>
<tr><td>Bit6</td><td>8 digits higher than the actual weight</td></tr>
<tr><td>Bit7</td><td>8 bits lower than actual weight</td></tr>
<tr><td>Bit8</td><td>Checksum high 8 bits</td><td rowspan="2">Check code= (bit2+bit3+....+bit6+bit7)</td></tr>
<tr><td>Bit9</td><td>Checksum low 8 bits</td></tr>
<tr><td>Bit10</td><td>End bit</td><td>0XFF (fixed)</td></tr>
</table>

Warm tip: when sending passively and actively, the operation and working mode 1 are the same, but the instructions are different;

Answers to questions about working mode 2:

Question 1: peeling operation is understandable. What's the matter with correction? Didn't you correct it when you delivered the goods?

A: friends who have contacted pressure sensors know that each pressure sensor needs to be recalibrated when used for the first time. The parameters of each sensor may not be exactly the same (for example, for sensors with a 5kg range, some coefficients K (which I personally understand as sensor linear curvature) are 430, some are 433, and some are 420). If it is not corrected, there will be errors in use, and individual sensors are even large （ Continue to look down, there are calibration methods below)

Question 2: bit4 displays positive and negative data representing weight. Is there any negative weight

Answer:

[step ①] weight_ Maopi = HX711_ Read();// Collect the ad value of fur weight once in advance

[step ②] weight_ Shiwu = HX711_ Read(); // Collect the ad value of the current weighing,

[step ③] weight_ jingzhong = Weight_ Shiwu -Weight_ Maopi;// The difference of subtraction is equal to the net weight ad value

[step ④] actual weight = weight_ jingzhong/K； // K is a coefficient, which is equivalent to the weight value returned under the command of channel a working mode 1, not the ad value. It has been converted to actual weight.

As described above, obtain net weight = measured weight - fur weight. When the stored fur weight is greater than the measured weight,

It will form a load-bearing state. The following 2 conditions can lead to weight bearing.

1: Man made to get the pressure sensor tray, giving it an upward force.

2: The error of the sensor fluctuates up and down in the fur weight value. Within a certain fluctuation range, this is normal.

It can be powered on for 2S, and then peeled after the voltage is stable to reduce the error; Or when the negative value is not large, ignore it handle. This depends on the functional requirements of the product designed by yourself, depending on the situation!

Correction method:

[Note: the correction is only effective for channel a, and channel B only has the peeling and zeroing function, without this correction function,]

Fur and correction parameter k are power-off saving functions, which can be operated once.

1: When entering the channel a correction function program, you must first send a 0xa5 command to the module. At this time, the calibration gate of channel a is opened, and the following calibration instructions are valid. After the correction is completed, you only need to send the required working mode (A1 / A2 / A3 / A4 / B1 / B2 / B3 / B4) command, and the AB channel correction gate function will be closed immediately. If you need to recalibrate, you will send the 0XA5 command again.

Send command to module	Function
0XAA	Conduct peeling (zero calibration and permanent Zeroing) (if working mode 2 is used, peeling must be carried out after the sensor and the module are connected.)
0XAB	The display weight is too large and needs to be reduced （ Actual weight = weight_ jingzhong/K;
0XAC	The display weight is too small and needs to be increased （ Actual weight = weight_ jingzhong/K;

The default unit is g. If the sensor with large range is accurate to g, the fluctuation will be large. It is recommended to take 0.1kg or 1kg. Equivalent to the output value divided by 10 or 100 or 1000

Since 0xab or 0xac is sent once, the correction coefficient K increases and decreases by 1, which is slow （ The value range of K is 1 to 999). Note: every time a correction instruction is sent, the module will return the corrected data, so as to check whether the correction is completed!

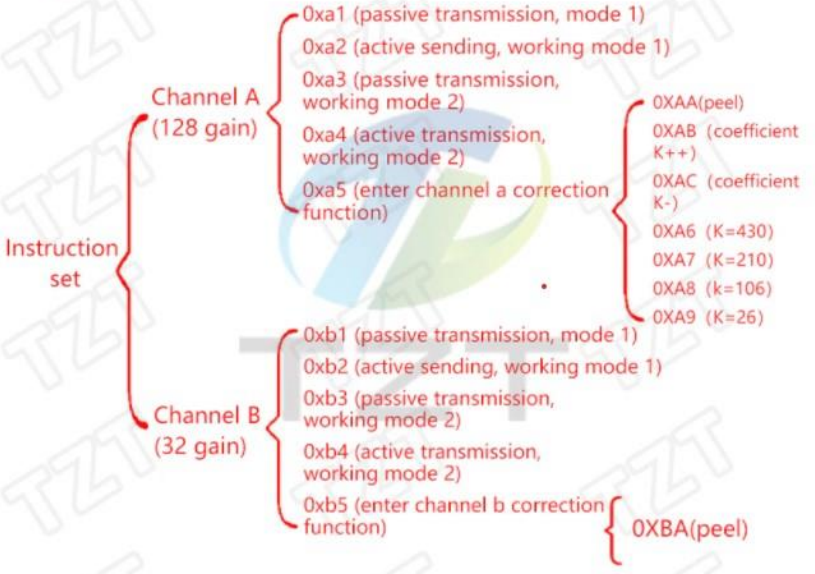
Here we add some commonly used K coefficient values (Note: the K value may be different for different manufacturers and batches, which need to be fine tuned and corrected), Directly sending this instruction can jump to the corresponding data, and then fine tune the correction by sending 0xab or 0xac plus or minus 1.

Fast correction send command	Corresponding parameters
0XA6	This code is a common 5kg range sensor, k = 430;
0XA7	This code is a common 10kg range sensor, k = 210;
0XA8	This code is a common 20kg or 40kg range sensor, k = 106;
0XA9	This code is a common 100kg or 200kg range sensor, k = 26 （ For 100kg of different models, the corresponding coefficient is 50. Test it when you use it. The coefficient is only for reference!)

Take the coefficient of 5kg range as an example: if the range you use is 5T (ton), according to the formula "actual weight = weight"_ The "Jingzhong / K" coefficient is taken as 430 of 5kg, and the weight remains unchanged_ The net weight ad value of Jingzhong is also unchanged. The actual weight range is from 5kg to 5T, which is enlarged by 1000 times. Then the obtained unit will be enlarged from G to 1000g. Then you get it. The coefficient of 5t range can also be 430, in kg; And so on, other ranges

2: When entering the B-channel correction function program, you must first send a 0xb5 command to the module. At this time, the calibration gate of channel B is opened, and then the peeling (zero calibration, permanent zero return) command 0xba is sent to be effective. After the operation is completed, only the required working mode (A1 / A2 / A3 / A4 / B1 / B2 / B3 / B4) command needs to be sent, and the AB channel correction gate function will be closed immediately. It is basically consistent with the operation of correcting channel a, but there is no function to modify the coefficient K.

Appendix 1: instruction set



Appendix 2: key points of hx711ad chip

I won't talk about how to use the hx711ad module here. For unclear information, please see the official data manual of hx711. Here I summarize the following points:

① : the module is divided into two groups of acquisition ports: channel A and channel B. The programmable gain of channel a is large, 128 or 64. The full-scale differential input voltage corresponding to these gains is ± 20mV or ± 40mv respectively. Channel B is a fixed 327 gain, and the corresponding full-scale differential input voltage is ± 80mv. Channel B corresponds to system parameter detection including battery.

Generally, the signal of pressure sensor is small. We use 128 gain of channel a. 64 gain is almost useless. Therefore, channel a of our serial port module is fixed at 128 gain, and channel B has only one gain, that is, 32 gain.

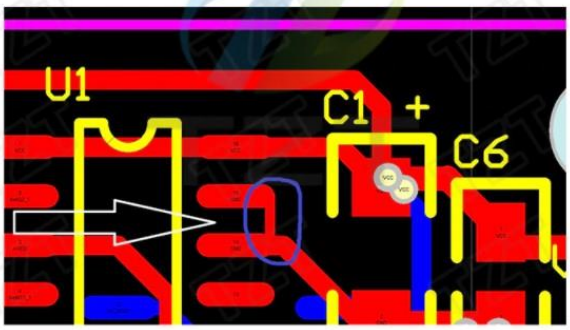
② : the internal crystal oscillator frequency of hx711 is divided into 10Hz and 80Hz. The accuracy of 10Hz is 1.5 bits higher than that of 80Hz. Because the data collected by the electronic scale in life is not so strict on time, 10 times in one second is also enough. Moreover, the refresh speed of the screen is too fast for your eyes to see! Therefore, the active mode of AB channel is transmitted every 100ms. If the passive mode (i.e. one sending and one receiving) is adopted, the interval between sending acquisition instructions must be greater than 100ms （ If customers require very fast acquisition speed, they can change the module to 80Hz working sequence. The interval between sending acquisition instructions in passive mode can reach 12.5ms, and in active mode, it is still sent once in 100ms. The frequency accuracy of 80Hz will be 1.5 bits lower than that of 10Hz, which is not recommended. See Appendix 3 for specific modification methods

③ : hx711 stipulates that there will be a stabilizer with 4 data cycles after a and B switch channels. In order to eliminate the error caused by switching channels as much as possible, a data cycle is added to our software. Therefore, 5 data cycles are delayed after channel switching. In the passive transmission mode, that is, after switching the channel, the first instruction needs to be executed for 500ms (sending the instruction again within 500ms is invalid). The second instruction in the same channel will be resumed to execute once in 100ms. If A and B output continuously and alternately, the instruction time interval needs to be at least 500ms (see Appendix III at the end of the document for the effect diagram). Compared with this case, the time is indeed longer. ① : it can be considered to change the working frequency of hx711 to 80Hz. A. B. the continuous alternating output time can be reduced to 62.5ms. ② : channel B is generally used to collect battery voltage status or other sensors with low time requirements. You can collect channel a 1000 times and then only collect channel B once to solve the problem.

Appendix 3: how to change the frequency of 10Hz to 80Hz?

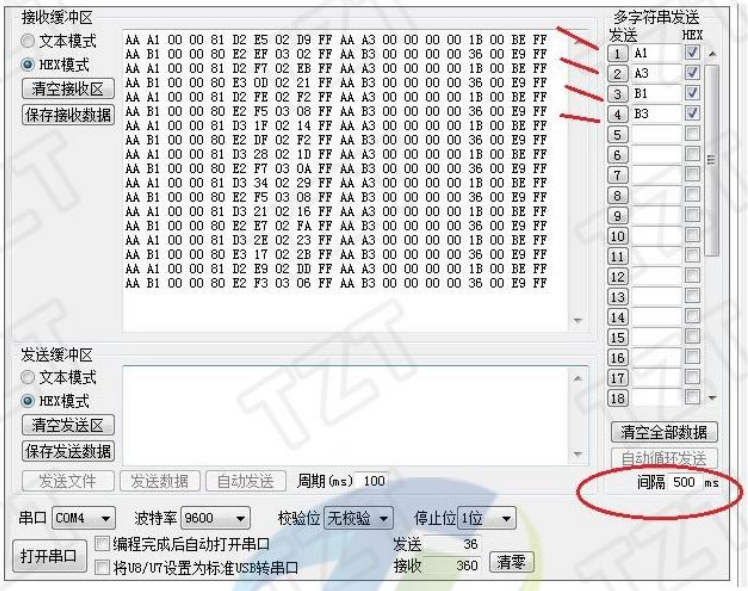
Modification method:

It is clearly written in the official data manual of hx711 that the 15th pin rate of hx711 is output rate control. 0:10hz 1:80hz, just connect this pin to the positive VCC of the power supply.

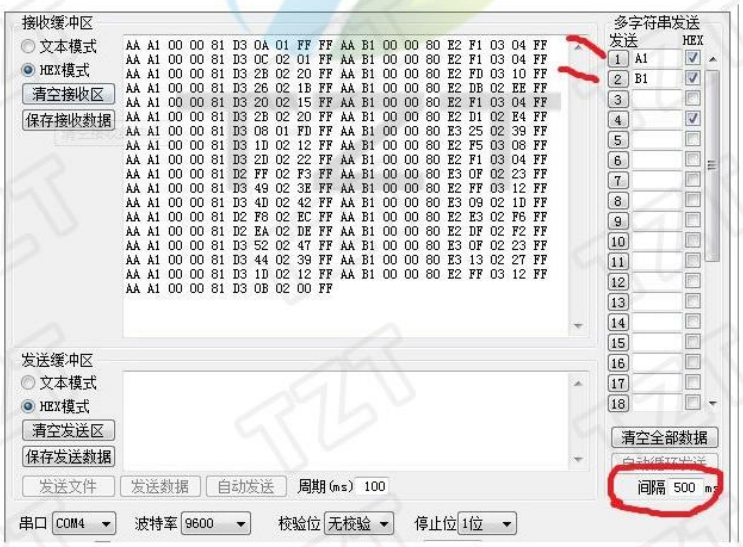


Appendix 4: continuous alternating output effect of AB channel

A1A3B1B3 Instruction alternate output



A1B1 Instruction alternate output



Appendix 5: 51 single chip microcomputer reference routine

The factory default is channel a active transmission; Working mode 1

Question answer: I use working mode 1 to collect the data of channel a or B, or working mode 2 to collect the difference ad of channel B. Why does data bit7 fluctuate so much?

A: first of all, we need to know that the module has a 24 bit resolution. If the ad invalid bits are not considered, the maximum data can reach the power of 2 ^ 24, which is equal to 16777216. Bit7 is only the lower 8 bits of the data. Even if bit7 is rounded off, there are 65536 resolutions, which are generally enough. So bit7 fluctuation is normal. Just like an electronic thermometer, if the resolution is accurate to 0.1 degrees, you may not see data fluctuations. However, if the resolution is accurate to 0.0001 degrees, the data will fluctuate rapidly all the time, such as 12.2568 ℃, 12.2658 ℃, 12.2902 ℃, 12.2056 ℃, etc. This is the subtle change after the sensor is amplified.

The program reading part of 51 single chip microcomputer is given as a gift. The complete program can view the program files in this compressed package.

```
/*********************************************************/
//Serial port interrupt service program
/*********************************************************/
void UartInt(void) interrupt 4
{
ES = 0; // Turn off serial port interrupt
while(! RI); // Byte1 fixed start bit 0XAA
Byte1=SBUF;
RI=0;
while(! RI); // Byte2 instruction return
Byte2=SBUF;
RI=0;
while(! RI); // Byte3 reserved
Byte3=SBUF;
RI=0;
while(! RI); // Byte4 reserved
Byte4=SBUF;
RI=0;
while(! RI); // Byte5 current channel low 16 bits
Byte5=SBUF;
RI=0;
while(! RI); // Byte6 current channel high 8 bits
Byte6=SBUF;
RI=0;
while(!RI); // Byte7 current channel lower 8 bits
Byte7=SBUF;
RI=0;
while(!RI); // Byte8 check bit high 8 bits
Byte8=SBUF;
RI=0;
while(!RI); // Byte9 check bit lower 8 bits
Byte9=SBUF;
RI=0;
```

RI=0;

while(!RI); // Byte10 fixed end bit 0xff
Byte10=SBUF;
RI=0;

If (byte1 = = 0XAA & & byte10 = = 0xff) / / judge the first byte and the last byte

{

If ((byte8 * 256 + byte9) = = (byte2 + byte3 + byte4 + byte5 + byte6 + byte7)) / / verify whether the check digit is correct

{

If (byte2 = = 0xa2) / / you can add a to determine whether to return the corresponding current instruction. If accurate, perform the following procedure

{

Weight=Byte5*65536+Byte6*256+Byte7; // Calculate the detection result (the A2 returned here is the ad value, and restore the value of hx711 [000000 h-ffffh])

}

}

}

ES = 1; // Enable serial port interrupt

Appendix 6: can it be networked through 232 / 485?

A: the communication of the blue serial port module is one-to-one. It can't be connected to the bus. If you need more than one network, please contact us at our after-sales QQ: 562409206. Provide low-cost networking solutions, and one bus can be attached to more than 200. Each pressure sensor can be peeled, zeroed and corrected remotely. The address bit of the pressure sensor can be set through the serial port between [00 ~ FF].

Hx711 Blue Board (AD Serial Port Module)