SE-8008 Sensor extender Datasheet

Introduce

The SE-8008 is a sensor expansion board based on MassDuino , the main chip is MD-8088. MD-8088 integrates a wealth of digital peripherals and analog peripherals, it offers great flexibility and play space for the users of the product design. SE -8008 is using SPI to communicate with main processor, it can save a lot of ports, so that we can use the SE-8008 to extend more different sensors from the main system. If you have a need, in this way can also extend other modules from the control board, such as the LE-1600 , RC-1406(Mortor Driver), ADC acquisition module and so on. MD-8088 chip is editable in the Arduino environment, so we can use a USB to serial cable for SE-8008 for upgrade firmware. The module is high performance low power, low cost , and it support secondary development.

About MassDuino , please refer to: http://www.inhaos.com/uploadfile/otherpic/UM-MASSDUINO-V01-EN.pdf

Features

- Can be program in the Arduino environment, very easy to use
- Support secondary development
- 8K bytes of in-system programmable FLASH, innovative data encryption technology
- 504 bytes of data FLASH, support byte read (simulate E2PROM)
- 1K SRAM bytes on-chip
- Programmable synchronous / asynchronous USART
- Can work in master / slave mode SPI Serial Interface
- I2C -compatible two-wire serial communication interface protocols , supporting master and slave device mode
- 32 8it general purpose working registers
- Support the expansion of the chip in-circuit debugging functions
- By SWD interface programming for FLASH, EEPROM, system configuration area , the ISP function
- Up to 30 programmable I / O
- High-performance, low -power and low-cost

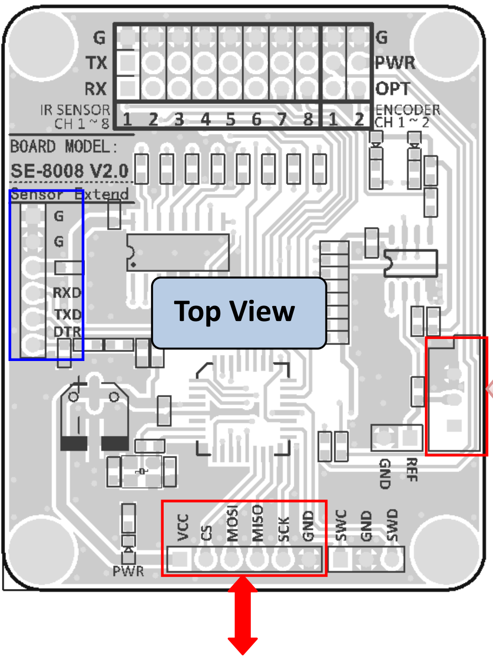
SPI communication ports

When we use the SE-8008, we can use the DuPong cable to connect it to the master chip SPI communication port

	SE-8008	Arduino UNO R3	Description
Pin Connection	VCC	VCC/3.3V	Operating Voltage
	CS	IO8	Chip Select Port
	MOSI	D11	SPI Interface
	MISO	D12	
	SCK	D13	
	GND	GND	GND

- **Pin Description**

MassDuino upload port , upload sketch via Arduino USB2SERIAL Light cable



Optical interrupter reference voltage adjust. since the optical interrupter output a analog signal , onboard comparator will shaping the signal to a digital signal , this VR used to adjust the reference level of the comparator

SPI Interface

- **SPI Protocol**

The LE-1600 was working in SPI Slave mode , after power on , the chip will wait for command form SPI Master ,then implement it if the it received valid command.

The communication protocol as below:

Package Format:				
Name	CMD ID	PARA_LEN	PARA_0 TO PARA_19	CKS
No. of the Bytes	0	1	2 ~ 21	22
Length (Bytes)	1	1	20	1
Value Range	0~255	0~255	0~255	0~255
Description	CMD type	CMD Valid length	Parameter, fill 0x00 for the invalid data	Bitwise NOT operation for sum of byte0 to byte6

In the master side , the SPI initial code as below:

```
//SPI Master side , to communication to LE-6000
#include <SPI.h>

int SE-8008_SS = 10; // assign SPI CS pin, in this code the SPI D8 connect to LE-1600’s SS Pin

void setup (void)
{
   pinMode(SE-8008_SS, OUTPUT);            // set SS pin is output
   digitalWrite(SE-8008_SS, HIGH);         // set SS pin to high , pull low the SS pin during SPI communication
   SPI.setClockDivider(SPI_CLOCK_DIV8);   // set SPI clock to 2MHz
   SPI.begin ();        // Start SPI communication
}
```

SE-8008 supported command as below:

Direction	Description	CMD ID	PARA_LEN	PARAMETER	Description
M -> S	READ_SENSOR_IR_OFF	0xDE	0		
M <- S	MSG_SENSOR_IR_OFF	0xDE	20	PARA_0: Value_H_#1 PARA_1: Value_L_#1 PARA_2: Value_H_#2 PARA_3: Value_L_#2 PARA_4: Value_H_#3 PARA_5: Value_L_#3 PARA_6: Value_H_#4 PARA_7: Value_7_#4 PARA_8: Value_H_#5 PARA_9: Value_L_#5 PARA_10: Value_H_#6 PARA_11: Value_L_#6 PARA_12: Value_H_#7 PARA_13: Value_L_#7 PARA_14: Value_H_#8 PARA_15: Value_L_#8 PARA_16: Value_OPT_H_#1 PARA_17: Value_OPT_L_#1 PARA_18: Value_OPT_H_#2 PARA_19: Value_OPT_H_#2	The data during IR OFF
M -> S	READ_SENSOR_IR_ON	0xDD	0		
M <- S	MSG_SENSOR_IR_ON	0xDD	20	SAME AS COMMAND 0XDE	The data during IR ON
M -> S	READ_SENSOR_DIFF	0xDC	0		
M <- S	MSG_SENSOR_DIFF	0xDC	20	SAME AS COMMAND 0XDE	The difference data between IR ON and OFF

- **IR Sensor Connection**

Below figure show the connection between IR sensor and SE-8008, they are connected via 3 wires: TX / RX and GND.

SENSOR_IR_OFF: the TX will set to "0", the IR will turned off , then read RX by 10bit ADC , this is the background IR signal.

SENSOR_IR_ON: the TX will set to "1", the IR will turned ON, then read RX by 10bit ADC , the signal level depend on the distance of obstacles.

SENSOR_IR_DIFF: the difference between SENSOR_IR_OFF and SENSOR_IR_ON. most IR detector use comparator to shaping the signal with a fixed reference level, when we use IR to detect distance of obstacles , the result is very depends on the environment , for the example , some time the unit is working very good in room , but when you turned light or move to sunlit environment, it will occurred unexpected error , because the light or sunshine have very high IR signal , this will change the background level and make the IR detector working failed. In this case , you can read SENSOR_IR_DIFF , this value will significantly reduce the environment changes impact the detection result.

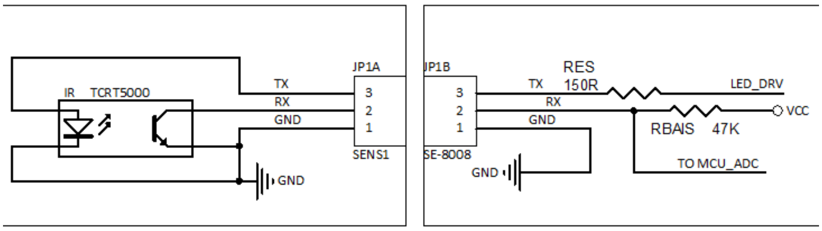
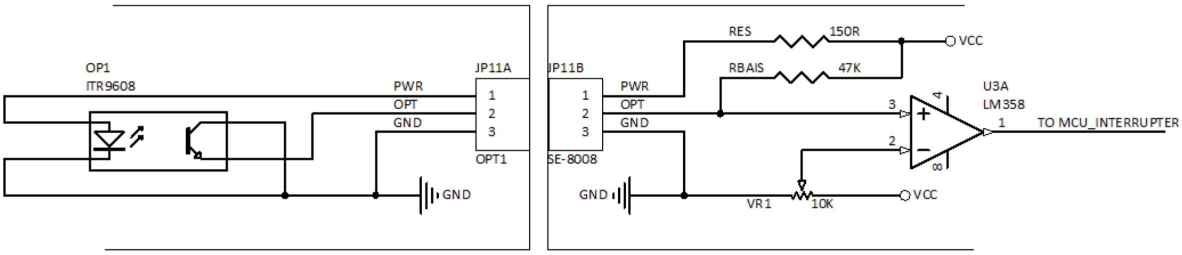


Photo interrupter Connection

The SE-8008 include 8-CH of IR detection and 2-CH of Photo interrupter, the IR detection used to detect distance of obstacles and photo interrupter used to detect wheel rotation speed, Below figure show the photo interrupter connection and how it work.



Difference with the IR detection circuit, the IR is always powered , and the OPT signal will shaping by a comparator, thinking difference application have difference signal level , so adjust the 10K VR to adjust reference level of the comparator.

The photo interrupter will not impact by the environment changes.

Arduino Source Code

Two code will coming with this board:

1, SE-8008.rar

This is arduino source code for SE-8008 , to development the SE-8008 , user need to install MassDuino driver, and write code in arduino , and upload sketch via USB2SERIAL light cable, the relational link as below:

MassDuino support package: http://www.inhaos.com/downcount.php?download_id=139

USB2SERIAL Light cable: http://www.inhaos.com/uploadfile/otherpic/DOC-BUONO-USB2SERIAL-V01-EN.pdf

2, SE-8008_UNO_MASTER.rar

This is arduino source code for UNO , the UNO will working with SE-8008, this code demo how to use SE-8008 in Arduino.

All code can be download from www.inhaos.com

Application:

- Extend IO and save code for UNO to drive many sensors
- Arduino car / robotics
- Sensor drive
- Toy

Contact us

1111 Oakmont Drive #C, San Jose, CA 95117
Contact: John Huang
Tel No: +1-408-981-6615
E-mail: support@inhaos.com
Website: www.inhaos.com