**调试记录**

7月25日

应用程序调试记录如下：

取其中一部分：

uthor: Gao Ming <sensarliar@gmail.com> 2014-07-25 17:20:10

Committer: Gao Ming <sensarliar@gmail.com> 2014-07-25 17:20:10

Parent: 766140f8f7317b4067c837558adb9c1361ef2d92 (add dvi infomation,add 3 speed)

Child: f291937a5b4eb161cd1faaf0556c7eaba5c15ab6 (nothing added)

Branches: gps\_parse\_usb, remotes/origin/gps\_parse\_usb

Follows:

Precedes:

add novatel receiver,add 3 direction speed

--------------------------------- gps\_nmea.cpp ---------------------------------

index 71d46c3..8867e32 100644

@@ -370,6 +370,136 @@ void parse\_nmea\_NVDVI(void) {

+

+/\*\*

+ \* parse BESTVELA, NOVATEL messages stored in

+ \* gps\_nmea.msg\_buf .

+ \* #BESTVELA,COM1,0,68.5,FINESTEERING,1802,459952.900,00000000,8c02,12832;SOL\_COMPUTED,DOPPLER\_VELOCITY,0.150,0.000,0.0505,12.520238,-0.0537,0.0\*18021b75

+ \*/

+void parse\_novatel\_bestvela(void) {

+ int i = 9; // current position in the message, start after: bestvela,

+ char\* endptr; // end of parsed substrings

+

+

+ // attempt to reject empty packets right away

+ if(gps\_nmea.msg\_buf[i]==',' && gps\_nmea.msg\_buf[i+1]==',') {

+ NMEA\_PRINT("p\_GPGGA() - skipping empty message\n\r");

+ return;

+ }

+

+

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field: COM1

+ if (i >= gps\_nmea.msg\_len)

+ return;

+ }

+

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field: 0

+ if (i >= gps\_nmea.msg\_len)

+ return;

+ }

+

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field: 68.5

+ if (i >= gps\_nmea.msg\_len)

+ return;

+ }

+

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field:FINESTEERING

+ if (i >= gps\_nmea.msg\_len)

+ return;

+ }

+

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field:1802

+ if (i >= gps\_nmea.msg\_len)

+ return;

+ }

+

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field:459952.900

+ if (i >= gps\_nmea.msg\_len)

+ return;

+ }

+

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field:00000000

+ if (i >= gps\_nmea.msg\_len)

+ return;

+ }

+

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field:8c02

+ if (i >= gps\_nmea.msg\_len)

+ return;

+ }

+

+ while(gps\_nmea.msg\_buf[i++] != ';') { // next field:12832;;;

+ if (i >= gps\_nmea.msg\_len)

+ return;

+ }

+

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field:SOL\_COMPUTED

+ if (i >= gps\_nmea.msg\_len)

+ return;

+ }

+

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field:DOPPLER\_VELOCITY

+ if (i >= gps\_nmea.msg\_len)

+ return;

+ }

+

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field:latency，0.150

+ if (i >= gps\_nmea.msg\_len)

+ return;

+ }

+

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field:Differential age in seconds，0.000

+ if (i >= gps\_nmea.msg\_len)

+ return;

+ }

+

+

+

+ int j=0;

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field: horizontal speed

+ if (i >= gps\_nmea.msg\_len) {

+ NMEA\_PRINT("p\_GPGGA() - skipping incomplete message\n\r");

+ return;

+ }

+ gps.speed\_horizon\_ch[j++]=gps\_nmea.msg\_buf[i-1];

+ }

+ gps.speed\_horizon\_ch[j]='\0';

+

+ double speed\_H = strtod(&gps.speed\_horizon\_ch[0],&endptr);

+

+ j=0;

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field: speed angle

+ if (i >= gps\_nmea.msg\_len) {

+ NMEA\_PRINT("p\_GPGGA() - skipping incomplete message\n\r");

+ return;

+ }

+ gps.speed\_angle\_ch[j++]=gps\_nmea.msg\_buf[i-1];

+ }

+ gps.speed\_angle\_ch[j]='\0';

+

+ double speed\_angle = strtod(&gps.speed\_angle\_ch[0],&endptr);

+

+ gps.speed\_E= speed\_H\*sin(speed\_angle);

+ gps.speed\_N= speed\_H\*cos(speed\_angle);

+

+

+ j=0;

+ while(gps\_nmea.msg\_buf[i++] != ',') { // next field: vertical UP speed

+ if (i >= gps\_nmea.msg\_len) {

+ NMEA\_PRINT("p\_GPGGA() - skipping incomplete message\n\r");

+ return;

+ }

+ gps.speed\_U\_ch[j++]=gps\_nmea.msg\_buf[i-1];

+ }

+ gps.speed\_U\_ch[j]='\0';

+

+

+

+

+}

+

+

+

/\*\*

\* parse\_nmea\_char() has a complete line.

\* Find out what type of message it is and

@@ -378,25 +508,26 @@ void parse\_nmea\_NVDVI(void) {

void nmea\_parse\_msg( void ) {

- if(gps\_nmea.msg\_len > 5 && !strncmp(gps\_nmea.msg\_buf , "NVGGA", 5)) {

+ if(gps\_nmea.msg\_len > 5 && !strncmp(gps\_nmea.msg\_buf , "GPGGA", 5)) {

gps\_nmea.msg\_buf[gps\_nmea.msg\_len] = 0;

NMEA\_PRINT("parse\_gps\_msg() - parsing GGA gps-message \"%s\" \n\r",gps\_nmea.msg\_buf);

NMEA\_PRINT("GGA");

parse\_nmea\_GPGGA();

}

else {

- if(gps\_nmea.msg\_len > 5 && !strncmp(gps\_nmea.msg\_buf , "NVVTG", 5)) {

+ if(gps\_nmea.msg\_len > 5 && !strncmp(gps\_nmea.msg\_buf , "GPVTG", 5)) {

gps\_nmea.msg\_buf[gps\_nmea.msg\_len] = 0;

NMEA\_PRINT("GSA: \"%s\" \n\r",gps\_nmea.msg\_buf);

NMEA\_PRINT("GSA");

parse\_nmea\_NVVTG();

} else {

- if(gps\_nmea.msg\_len > 5 && !strncmp(gps\_nmea.msg\_buf , "NVDVI", 5)) {

+ if(gps\_nmea.msg\_len > 5 && !strncmp(gps\_nmea.msg\_buf , "BESTVELA", 8)) {

gps\_nmea.msg\_buf[gps\_nmea.msg\_len] = 0;

NMEA\_PRINT("GSA: \"%s\" \n\r",gps\_nmea.msg\_buf);

NMEA\_PRINT("GSA");

- parse\_nmea\_NVDVI();

+ //parse\_nmea\_NVDVI();

+ parse\_novatel\_bestvela();

// parse\_nmea\_GPGSA();

} else {

gps\_nmea.msg\_buf[gps\_nmea.msg\_len] = 0;

@@ -420,7 +551,7 @@ void nmea\_parse\_msg( void ) {

void nmea\_parse\_char( char c ) {

//reject empty lines

if (gps\_nmea.msg\_len == 0) {

- if (c == '\r' || c == '\n' || c == '$')

+ if (c == '\r' || c == '\n' || c == '#'|| c == '$')

return;

}

---------------------------------- gps\_nmea.h ----------------------------------

index 08575f0..20154d0 100644

@@ -60,10 +60,15 @@ struct GpsState {

char direction\_ch[12];

char speed\_ch[12];

+ char speed\_horizon\_ch[12];

+ char speed\_angle\_ch[12];

char speed\_E\_ch[12];

char speed\_N\_ch[12];

char speed\_U\_ch[12];

+ double speed\_E;

+ double speed\_N;

+

/\*

int16\_t gspeed; ///< norm of 2d ground speed in cm/s

int16\_t speed\_3d; ///< norm of 3d speed in cm/s

------------------------------ gps\_test.pro.user ------------------------------

index 1c02544..9a3e39b 100644

@@ -1,6 +1,6 @@

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE QtCreatorProject>

-<!-- Written by QtCreator 2.7.1, 2014-07-16T09:26:04. -->

+<!-- Written by QtCreator 2.7.1, 2014-07-25T17:16:35. -->

<qtcreator>

<data>

<variable>ProjectExplorer.Project.ActiveTarget</variable>

-------------------------------- mainwindow.cpp --------------------------------

index 3c96626..126d30f 100644

@@ -353,6 +353,10 @@ void MainWindow::remoteDataIncoming()

ui->m\_direction->display((char \*)gps.direction\_ch);

//ui->m\_hight->display(temp\_value.setNum(gps.alt));

ui->m\_hight->display((char \*)gps.alt\_ch);

+

+ ui->m\_speed\_U->display((char \*)gps.speed\_U\_ch);

+ ui->m\_speed\_E->display(gps.speed\_E);

+ ui->m\_speed\_N->display(gps.speed\_N);

}

else{

ui->m\_label\_available->setText(QString("N/A"));

@@ -365,6 +369,9 @@ void MainWindow::remoteDataIncoming()

ui->m\_speed->display("");

ui->m\_direction->display("");

ui->m\_hight->display("");

+ ui->m\_speed\_U->display("");

+ ui->m\_speed\_E->display("");

+ ui->m\_speed\_N->display("");

}

//QString disp\_hight;

8月3日

把应用程序gps\_test\_0618\_1405拷贝到/home/root的目录下：

制作ubi文件系统的命令如下，采用小型文件系统：

./mkfs.ubifs -r .././arago\_fs\_nogit/ -F -o ubifs.img -m 2048 -e 126976 -c 3992

./ubinize -o ubi.img -m 2048 -p 128KiB -s 512 -O 2048 ubinize.cfg

这样就制作好了ubi.img 文件系统。

8月6日

把制作好的ubi.img考入到制作sd卡的文件夹的xxx/prebuilt-zfcs-mksdcard-git/boot的目录中，用

sudo ./mksdcard.sh --device /dev/sdb

命令烧写sd卡，

用mount命令确认sd卡使用usb或者其他挂载在/dev/sdb 上了；否则会损坏所操作的硬盘或磁盘。

8月8日：

烧写nand flash的步骤如下，

在sd卡上电启动的时候，立即按回车键，进入u-boot的启动界面：

然后输入如下命令行：

mw.b 0x82000000 0xFF 0x20000

mmc rescan

fatload mmc 0 0x82000000 MLO

nandecc hw 2

nand erase 0x0 0x20000

nand write 0x82000000 0x0 0x20000

mw.b 0x82000000 0xFF 0x1E0000

mmc rescan

fatload mmc 0 0x82000000 u-boot.img

nandecc hw 2

nand erase 0x80000 0x1E0000

nand write 0x82000000 0x80000 0x1E0000

saveenv

mw.b 0x82000000 0xFF 0x500000

mmc rescan

fatload mmc 0 0x82000000 uImage

nandecc hw 2

nand erase 0x280000 0x500000

nand write 0x82000000 0x280000 0x500000

///mw.b 0x82000000 0xFF 0x1F880000

mw.b 0x82000000 0xFF 0x3200000

mmc rescan

fatload mmc 0 0x82000000 ubi.img

nandecc hw 2

nand erase 0x780000 0x1F880000

//nand erase 0x780000 0x3200000

nand write 0x82000000 0x780000 ${filesize}

///nand write 0x82000000 0x280000 0x500000

就可以成功地把 程序 写入nand flash中了。

8月22日

焊接好后的裸板，需要进行如下的配置，ARM板才可以启动：

建议 ：

1,用sd卡中的filesystem制作ubi文件系统。

用cp命令的方式拷贝arago文件系统；这样就ok，文件系统可以工作。启动时间为40s。

log记录为/home/gaoming/log/test\_fs/gm\_ubi\_t3.txt

2,用uboot 命令行烧写nandflash

3,对nandflash的烧写，需要重新在板子上焊接电阻 ，配置启动顺序 。

把R66拆至R91;把R67拆至R92;boot3,boot4

从配置为11100 sd卡启动

改为00100, 启动顺序为MMC0->NAND

附录：

1)如果要加速可以用nand为第一启动顺序的配置；

nandflash最先启动可用的为 10010;10011;10100;

sd卡最先启动的为10111;

2)关掉网口启动。

8月28日：

修改设置板卡的命令记录：

对于com2口，

原始数据和星历保持3种数据

分别为：

板卡初始化的串口速率为9600，先设置com1为115200

serialconfig com1 115200 n 8 1 n off

log gpgga ontime 0.25

log gpvtg ontime 0.25

log bestvola ontime 0.25

log gpzda ontime 1

serialconfig com2 921600 n 8 1 n off

log com2 rangecmpb ontime 0.05

log com2 rawephemb ontime 300

log com2 bdsephemerisb ontime 300

saveconfig

数据接收并转换为asiic码如下所示：

#RANGECMPA,COM2,0,80.5,FINEBACKUPSTEERING,1809,216898.400,00000000,9691,12832;9,249d0008cf16fcaf1577240aca6a6fd6301deeb140030000,449d1008075c06502badcc09330b1a90400f76af00030000,4b3d300197f40400d4accc099793ecfb5b0f0a0000000000,849e04081cf8ff6f15565f11b59031b16009a99d80020000,a49e0408f3b905c08c1f891319fbbac8600a141380020000,c49f040830f1f8bfe988b9120cc5d9cfa108eb7000020000,e49f04084a1e0780260de91292dcebb07007f36460020000,049c84187cf1ff5f462b9f11b20ba587500339aac0020000,249d041855bffabf324d970a7d43319b600ce477a0020000\*ad4b444b

#RANGECMPA,COM2,0,80.5,FINEBACKUPSTEERING,1809,216898.450,00000000,9691,12832;9,249d0008a916fcdf6177240ab9386fd6301df0b140030000,449d1008d75b0660afaccc09995c1a90400f78af00030000,4b3d300172f4041058accc0905d3ecfb5b0f0c0000000000,849e04082cf8ff0f16565f11519031b16009aa9d80020000,a49e040816ba05201c1f89136644bbc8600a161380020000,c49f0408bff1f87f7489b912c16ad9cfa108ec7000020000,e49f0408271e07709a0ce912ad37ecb07007f56460020000,049c8418a4f1ff7f472b9f11fb0aa58750033aaac0020000,249d041832bffa0f9a4d970a3f00319b600ce677a0020000\*a11754d4

#RAWEPHEMA,COM2,4,70.5,FINEBACKUPSTEERING,1809,216600.000,00000000,58ba,12832;3,1805,93600,8b0c341d74a7c350fc5dc7b3ce2baed60a1c9d69f60f16da00002b34ba34,8b0c341d75290ff5df3b78dee74480f7cc08e10c0e04bfa10e8a6416da57,8b0c341d75afff9251ac45e3005626428d8e28e139a432c1ff9c520ff860\*7cef29b6

#RAWEPHEMA,COM2,3,75.5,FINEBACKUPSTEERING,1809,216600.000,00000000,58ba,12832;9,1809,180000,8b0c44394ea7c451fc14a3bdb0fa935b3672f23008642bf200fffe02ab40,8b0c44394ca864fdf038430364ef90fe0c0037754d055aa17033d12bf27f,8b0c44394d2dffdcc35ff68dffe82714fd16289491220004ffa2f664f7e9\*8c256cfc

#BDSEPHEMERISA,COM2,6,70.5,FINESTEERING,1809,216534.000,00000000,2626,12832;2,453,2.00,1,5.70e-09,-1.37e-08,0,201600,8.54127691e-04,7.46869233e-12,0.00000000,1,201600,6493.457827,3.5836501047e-04,-2.062648829,2.9140499533e-09,0.325050405,-1.014182762e+00,-2.04937108e-09,9.4532844644e-02,-7.393165098e-11,-2.627726644e-05,-1.894310117e-05,576.6718750,-795.6718750,7.124617696e-08,1.629814506e-08\*bee0db3f

#BDSEPHEMERISA,COM2,5,78.0,FINESTEERING,1809,216534.000,00000000,2626,12832;13,450,2.00,1,8.40e-09,3.20e-09,12,464400,2.90118274e-04,1.65423231e-11,1.16551734e-18,13,464400,5282.613491,2.4546941277e-03,-3.117404147,3.9208776060e-09,-1.877576954,9.5126697220e-01,-6.76171022e-09,9.5695315484e-01,-2.528676758e-10,9.015202522e-07,9.852461517e-06,157.7187500,16.5312500,-2.654269338e-08,-1.722946763e-08\*14657d54

9月3日：

例试过程：

对BDS/GPS定位授时设备进行功能试验。必要时，可根据试验需要，调整试验项目中高温试验、低温试验、振动试验顺序（GPS/BD天线需放置在空旷无遮挡处进行试验）。

1. 外观检查：产品外观整洁无损，标识清晰可辩，装配螺钉固定牢固，插座无松动，测量设备三维尺寸是否满足技术协议要求。
2. 常温试验

原理简述：根据试验线路图，向设备连接电源、天线、输出信号。电源采用28V直流电源供电；

测试过程：1) 连接好天线后，用示波器观察IRIG-B AC码、DC码和1PPS的输出波形是否正常；2) 观察显示屏的时间，位置，速度信息(测试状态为静止，速度为0)是否正确，U盘数据存储是否正确；3）观察时间解调板的信息时间是否和1588发送的时间戳一致。

3. 振动试验

按照按HB5830功能正弦谱E曲线对设备进行正弦扫描，测试方向为安装方向，试验过程中检查产品：LCD屏幕上时间位置信息显示正常，设备结构件无松动、接插件完好。

1. 电源波动

将直流电源波动至18V，检查设备LCD屏幕是否工作正常，若正常，再调整至32V，检查设备LCD屏幕是否工作正常。

5. 高温试验

将设备放置于高低温箱中，+70℃保温1小时后，检查设备LCD屏幕是否显示时间、位置信息，并根据常温试验中3项测试方法检查设备工作是否正常。

6. 低温试验

将设备放置于高低温箱中，-30℃保温1小时后，检查设备LCD屏幕是否显示时间、位置信息，并根据常温试验中3项测试方法检查设备工作是否正常。

例试摸底中出现的问题和注意事项：

1，低温-40度保持1个半小时后，需要3~5分钟才能启动。  
2，低温时，出现天线接口接触不好，重新插紧后恢复正常。（推断所用的蘑菇头天线时间太久，接口松动导致的，最好重新换好一点的天线。）

3，天线需放在空旷无遮挡的地方，修改程序，增加未定位之前的收星个数。

9月9日

1. 摸底测试sandisk 32G u盘，高温70度，低温-30度各测试一个小时，可以正常工作。

2） 在谢新处补测BDS/GPS授时定位系统支持32G u盘。

9月10日

优化启动时间需要做的事情：

1. 修改启动电阻，nand flash先启动，这样可以加快一点儿速度。
2. 用最小文件系统，这样也会加快一些速度。

9月17日

1)修改 加入 检测 未锁定前 的 GPS和 北斗 的卫星数量，供安装调试 用。

2)修改 界面，把字体变大 ，用图片代替无法改变大小的字体。

3)文件系统中可以把 不用的字体删掉，/usr/lib/fonts

9月19日

1)在上次已经定位一段时间后，上电启动时间

从SD卡启动时间： 30s，定位33s

2)烧写 nandflash后第一次启动log。

6.566402] sd 0:0:0:0: [sda] 8028160 512-byte logical blocks: (4.11 GB/3.82 GiB)

[ 6.584408] drivers/rtc/hctosys.c: unable to open rtc device (rtc0)

[ 6.591778] sd 0:0:0:0: [sda] Write Protect is off

[ 6.603405] sd 0:0:0:0: [sda] No Caching mode page present

[ 6.609191] sd 0:0:0:0: [sda] Assuming drive cache: write through

[ 6.667162] sd 0:0:0:0: [sda] No Caching mode page present

[ 6.672952] sd 0:0:0:0: [sda] Assuming drive cache: write through

[ 6.693378] sda: sda1

[ 6.706242] sd 0:0:0:0: [sda] No Caching mode page present

[ 6.712048] sd 0:0:0:0: [sda] Assuming drive cache: write through

[ 6.718519] sd 0:0:0:0: [sda] Attached SCSI removable disk

[ 6.795059] UBIFS: start fixing up free space

[ 11.548771] UBIFS: free space fixup complete

[ 11.624638] UBIFS: mounted UBI device 0, volume 0, name "rootfs"

[ 11.630993] UBIFS: file system size: 505491456 bytes (493644 KiB, 482 MiB, 3981 LEBs)

[ 11.639461] UBIFS: journal size: 9023488 bytes (8812 KiB, 8 MiB, 72 LEBs)

[ 11.647179] UBIFS: media format: w4/r0 (latest is w4/r0)

[ 11.653320] UBIFS: default compressor: lzo

[ 11.657643] UBIFS: reserved for root: 0 bytes (0 KiB)

[ 11.706544] VFS: Mounted root (ubifs filesystem) on device 0:13.

[ 11.713407] Freeing init memory: 236K

3) nand flash 第二次开始记录log。

6.240826] Detected MACID=0:18:31:e6:b7:1b

[ 6.246327] cpsw: Detected MACID = 00:18:31:e6:b7:1c

[ 6.252664] drivers/rtc/hctosys.c: unable to open rtc device (rtc0)

[ 6.325829] UBIFS: recovery needed

[ 6.483961] UBIFS: recovery completed

[ 6.487838] UBIFS: mounted UBI device 0, volume 0, name "rootfs"

[ 6.494189] UBIFS: file system size: 505491456 bytes (493644 KiB, 482 MiB, 3981 LEBs)

[ 6.502622] UBIFS: journal size: 9023488 bytes (8812 KiB, 8 MiB, 72 LEBs)

[ 6.510337] UBIFS: media format: w4/r0 (latest is w4/r0)

[ 6.516491] UBIFS: default compressor: lzo

[ 6.520803] UBIFS: reserved for root: 0 bytes (0 KiB)

[ 6.527408] VFS: Mounted root (ubifs filesystem) on device 0:13.

[ 6.534262] Freeing init memory: 236K

[ 8.277364] scsi 0:0:0:0: Direct-Access ATP Nano Vision 1100 PQ: 0 ANSI: 0 CCS

[ 8.645660] sd 0:0:0:0: [sda] 8028160 512-byte logical blocks: (4.11 GB/3.82 GiB)

[ 8.782581] sd 0:0:0:0: [sda] Write Protect is off

[ 9.034771] sd 0:0:0:0: [sda] No Caching mode page present

[ 9.040583] sd 0:0:0:0: [sda] Assuming drive cache: write through

[ 9.676474] sd 0:0:0:0: [sda] No Caching mode page present

[ 9.682277] sd 0:0:0:0: [sda] Assuming drive cache: write through

[ 10.141847] sda: sda1

[ 10.209894] sd 0:0:0:0: [sda] No Caching mode page present

[ 10.215775] sd 0:0:0:0: [sda] Assuming drive cache: write through

[ 10.222199] sd 0:0:0:0: [sda] Attached SCSI removable disk

[ 11.609695] net eth0: CPSW phy found : id is : 0x7c0f1

4)nandflash 启动时间为40s，定位时间也为40s。未调整启动电阻。

5)10:31:09 开始记录

测试到13:50,共记录87MB。

CPU占用率为15%左右。

没个小时记录数据为25MB，（由于在窗口测试，可见卫星数不多，故数据量可能会变大）

6)修改电阻，配置为10100,最先从nandflash启动，

启动时间为35s。