

Atmega1284P 版 INVADER データシールド仕様書

1 データ仕様

- ARTSAT1:INVADER 衛星の運用期間中に取得した 2014 年 2 月 28 日 18 時 0 分 11 秒から同年 8 月 31 日 6 時 38 分 43 秒 (いずれも世界標準時) のログデータにより構成されている。

(1) レコード数 : 3518

(2) データ形式 (項目数 28)

	項目	整数部桁数	小数部桁数	書式	最小値	最大値
1	月	2	0	99	2	8
2	日	2	0	99	1	31
3	時	2	0	99	0	23
4	分	2	0	99	0	59
5	秒	2	0	99	0	59
6	太陽電池電流-Y2	3	1	999.9	0	250.9
7	太陽電池電流+Y2	3	1	999.9	0	209.1
8	太陽電池電流-Z	3	1	999.9	0	374.7
9	太陽電池電流+Z	3	1	999.9	0	577.2
10	太陽電池電流-Y1	3	1	999.9	0	193.0
11	太陽電池電流+Y1	3	1	999.9	0	318.9
12	太陽電池電流-X	3	1	999.9	0	418.3
13	太陽電池電流+X	3	1	999.9	0	450.9
14	電池温度 1	2	0	s99	-8	28
15	電池温度 2	2	0	s99	-8	56
16	電池温度 3	2	0	s99	-7	64
17	太陽電池温度-X	2	0	s99	-38	27
18	太陽電池温度+Y1	2	0	s99	-38	27
19	太陽電池温度+Y2	2	0	s99	-38	69
20	太陽電池温度-Y1	2	0	s99	-38	41
21	太陽電池温度+Z1	2	0	s99	-38	56
22	太陽電池温度+Z2	2	0	s99	-38	56
23	太陽電池温度-Z1	2	0	s99	-38	32
24	太陽電池温度-Z2	2	0	s99	-38	32
25	Main OBC 温度	2	0	s99	-38	33
26	ジャイロ X	1	1	S9.9	-3.7	3.7
27	ジャイロ Y	1	1	S9.9	-3.7	3.7
28	ジャイロ Z	1	1	S9.9	-3.7	3.7

- 注) 1 各項目の間にデリミタとして「,」が挿入されている csv 形式である。
 2 「月日時分秒」はデータ取得の際の世界標準時である。
 3 各値は衛星からの受信データを基に、各項目の桁数以下を切り捨てたものである。
 4 書式欄の「9」は数字一桁、「s」は値がマイナスの場合にはマイナス記号が付くことを表す。整数部桁数にはマイナス記号は含まれていない。
 5 各数値の桁数は固定であり、上位桁がゼロの場合は「0」が付く。
 6 レコード区分のため、各レコード末尾には「改行 (CR : 0x0D)」が付く。

2 Arduino との通信方法

Arduino のソフトシリアル機能利用

(デジタル端子 4、7、8、12 をジャンパピンで選択)

データ転送速度 : 9600bps (送受信とも)

3 データ送出の仕様

- (1) 電源投入直後 (リセット直後) は先頭 (2 月 28 日 18 時 00 分 11 秒 (世界標準時)) から開始
- (2) データ送出の約 1 秒又は 5 秒を選択。電源投入直後 (リセット直後) の間隔はジャンパ (中央寄り Atmega1284P の 40 番ピン) により選択 (オープンで 1 秒、ショートで 5 秒)
- (3) 動作中に Arduino からのシリアルコマンドで送出間隔等を切り替え可能
- (4) 最終レコード (INVADER は 8 月 31 日 6 時 38 分 43 秒 (世界標準時)、気球ピギーバックは 7 時 49 分 45 秒 (日本標準時)) まで読み出した後は上記(1)の最初のレコードに戻る。

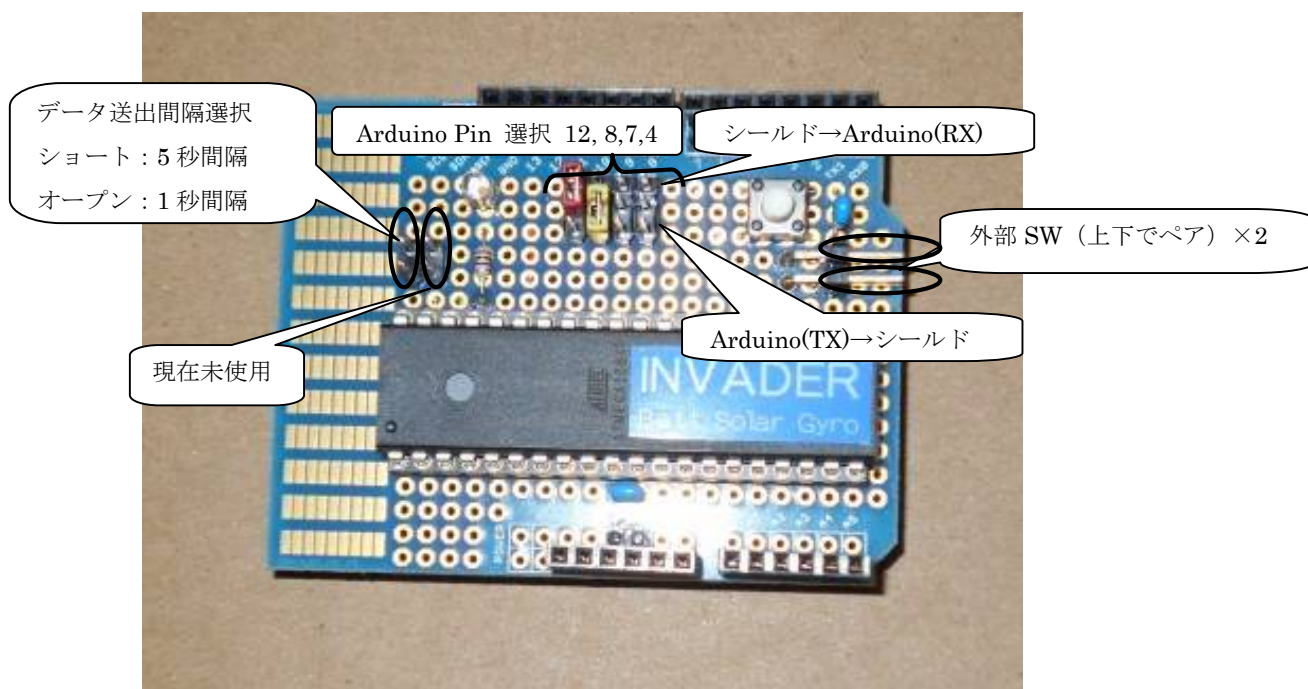
4 プッシュ・スイッチによるデータの送信・停止方法

- (1) シールド上のタクトスイッチを押すごとに送信・停止の交互切り替え
 - ・外部にプッシュスイッチ増設可能 (接続端子 (ピン) あり)
 - (ただし、ワンプッシュで確実にオンオフするにはさらに回路を追加する等の工夫が必要。接続するスイッチの性能にも左右される。)
- (2) データ送信中はシールド上の LED が点灯、停止で消灯。(コマンドによる送信及び停止時も同様)

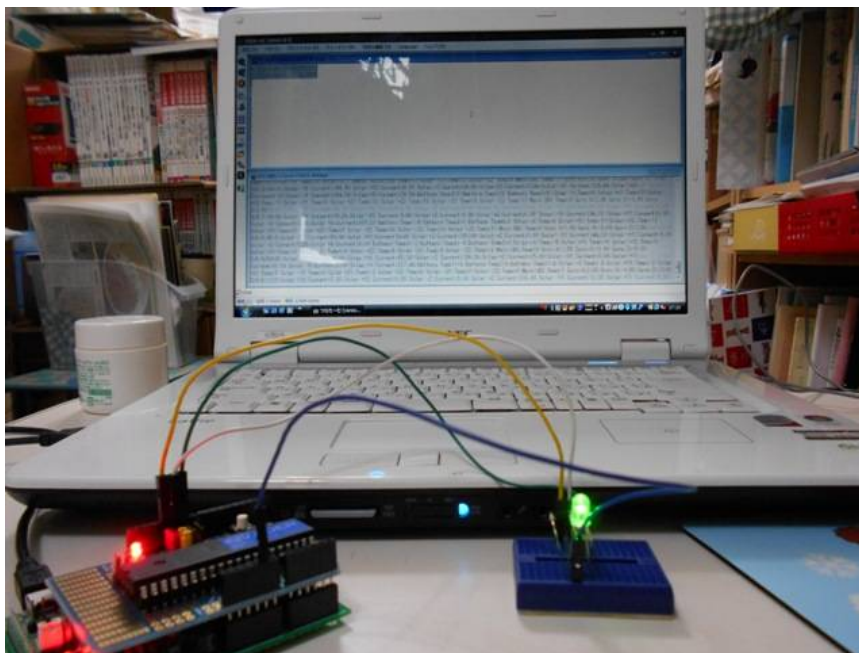
5 シリアル通信からの制御コマンド

コマンド	機能	機能詳細	留意事項
B	データ送信開始		直前の送信レコードの次、又は、送信停止時に設定されたレコード位置から、設定された送信間隔でデータ送信を開始する。
S	データ送信停止	データ送信の一時停止	
K	データ 1 レコード送信		※データ送信停止時有効
1	データ送信間隔 1 秒セット		直前の間隔で一度データ送信された後、送信間隔が変更される。
5	データ送信間隔 5 秒セット		直前の間隔で一度データ送信された後、送信間隔が変更される。
R	データ送信位置リセット		※データ送信停止時有効
P	データ送信位置設定	データ読出し位置を後に続く 4 又は 3 桁の数値（0000 ～ 3517）のレコード番号に設定	※データ送信停止時有効
M	データ送信位置 3 月設定	データ読出し位置を 3 月からに設定	※データ送信停止時有効
A	データ送信位置 4 月設定	データ読出し位置を 4 月からに設定	※データ送信停止時有効
Y	データ送信位置 5 月設定	データ読出し位置を 5 月からに設定	※データ送信停止時有効
J	データ送信位置 6 月設定	データ読出し位置を 6 月からに設定	※データ送信停止時有効
Y	データ送信位置 7 月設定	データ読出し位置を 7 月からに設定	※データ送信停止時有効
G	データ送信位置 8 月設定	データ読出し位置を 8 月からに設定	※データ送信停止時有効

6 シールド各ジャンパ線の説明



7 動作状況



動画掲載場所 <https://www.dropbox.com/s/0oqcwssggzj4gnu/DSCN0602.wmv?dl=0>

8 サンプルスケッチ

```

/*
  INVADER Shield

  Data from INVADER Shield are received from software serial.
  The data received from the shield are modified
  and sent to PC through hardware serial.
  The multicolor LED is lightened
  according to the value of the gyro data.
  The circuit:
  * RX is digital pin 12 (connect to TX of other device)
  * TX is digital pin 8 (connect to RX of other device)
  * LED red is connected to digital pin 9, blue is 10, green is 11 for example.

  modified 21 June 2015
  by Masahiro Sanada

  */
#include <SoftwareSerial.h>

const int  analogOutX  = 9;
const int  analogOutY  = 10;
const int  analogOutZ  = 11;
int  outXvalue  = 0;
int  outYvalue  = 0;
int  outZvalue  = 0;

SoftwareSerial mySerial(12, 8); // RX, TX

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  Serial.println("Hello, this is INVADER Shield.");

  // set the data rate for the SoftwareSerial port
  mySerial.begin(9600);
}

void loop()
{
  if (Serial.available()){
    mySerial.write(Serial.read());
    //You can send command through terminal software.
  }
  if (mySerial.available()){
    int  month = mySerial.parseInt();
    int  day = mySerial.parseInt();
    int  hour = mySerial.parseInt();
    int  minute = mySerial.parseInt();
    int  second = mySerial.parseInt();

    float  solarMinusY2  = mySerial.parseFloat();
    float  solarPlusY2   = mySerial.parseFloat();
    float  solarMinusZ   = mySerial.parseFloat();
    float  solarPlusZ    = mySerial.parseFloat();
    float  solarMinusY1  = mySerial.parseFloat();
    float  solarPlusY1   = mySerial.parseFloat();
    float  solarMinusX   = mySerial.parseFloat();
  }
}

```

```

float  solarPlusX  =  mySerial.parseFloat();

int  batTemp1  =  mySerial.parseInt();
int  batTemp2  =  mySerial.parseInt();
int  batTemp3  =  mySerial.parseInt();

int  solarTempMinusX  =  mySerial.parseInt();
int  solarTempPlusY1  =  mySerial.parseInt();
int  solarTempPlusY2  =  mySerial.parseInt();
int  solarTempMinusY1  =  mySerial.parseInt();
int  solarTempPlusZ1  =  mySerial.parseInt();
int  solarTempPlusZ2  =  mySerial.parseInt();
int  solarTempMinusZ1  =  mySerial.parseInt();
int  solarTempMinusZ2  =  mySerial.parseInt();

int  mainOBCTemp  =  mySerial.parseInt();

float  gyroX  =  mySerial.parseFloat();
float  gyroY  =  mySerial.parseFloat();
float  gyroZ  =  mySerial.parseFloat();

int  D  = mySerial.read();//read the rest of buffer

Serial.print(month);
Serial.print('/');
Serial.print(day);
Serial.print(' ');
Serial.print(hour);
Serial.print(':');
Serial.print(minute);
Serial.print(':');
Serial.print(second);
Serial.print(' ');
Serial.print("Solar -Y2 Current:");
Serial.print(solarMinusY2);
Serial.print(' ');
Serial.print("Solar +Y2 Current:");
Serial.print(solarPlusY2);
Serial.print(' ');
Serial.print("Solar -Z Current:");
Serial.print(solarMinusZ);
Serial.print(' ');
Serial.print("Solar +Z Current:");
Serial.print(solarPlusZ);
Serial.print(' ');
Serial.print("Solar -Y1 Current:");
Serial.print(solarMinusY1);
Serial.print(' ');
Serial.print("Solar +Y1 Current:");
Serial.print(solarPlusY1);
Serial.print(' ');
Serial.print("Solar -X Current:");
Serial.print(solarMinusX);
Serial.print(' ');
Serial.print("Solar +X Current:");
Serial.print(solarPlusX);
Serial.print(' ');
Serial.print("Battery Temp1:");
Serial.print(batTemp1);

```

```

Serial.print(' ');
Serial.print("Battery Temp2:");
Serial.print(batTemp2);
Serial.print(' ');
Serial.print("Battery Temp3:");
Serial.print(batTemp3);
Serial.print(' ');
Serial.print("Solar -X Temp:");
Serial.print(solarTempMinusX);
Serial.print(' ');
Serial.print("Solar +Y1 Temp:");
Serial.print(solarTempPlusY1);
Serial.print(' ');
Serial.print("Solar +Y2 Temp:");
Serial.print(solarTempPlusY2);
Serial.print(' ');
Serial.print("Solar -Y1 Temp:");
Serial.print(solarTempMinusY1);
Serial.print(' ');
Serial.print("Solar +Z1 Temp:");
Serial.print(solarTempPlusZ1);
Serial.print(' ');
Serial.print("Solar +Z2 Temp:");
Serial.print(solarTempPlusZ2);
Serial.print(' ');
Serial.print("Solar -Z1 Temp:");
Serial.print(solarTempMinusZ1);
Serial.print(' ');
Serial.print("Solar -Z2 Temp:");
Serial.print(solarTempMinusZ2);
Serial.print(' ');
Serial.print("Main OBC Temp:");
Serial.print(mainOBCTemp);
Serial.print(' ');
Serial.print("Gyro X:");
Serial.print(gyroX);
Serial.print(' ');
Serial.print("Gyro Y:");
Serial.print(gyroY);
Serial.print(' ');
Serial.print("Gyro Z:");
Serial.print(gyroZ);
Serial.print("¥r");

outXvalue = map(gyroX,-3.7,3.7,0,255);
outYvalue = map(gyroY,-3.7,3.7,0,255);
outZvalue = map(gyroZ,-3.7,3.7,0,255);
analogWrite (analogOutX,outXvalue);
analogWrite (analogOutY,outYvalue);
analogWrite (analogOutZ,outZvalue);

```

```

}

```

```

}

```