2015/07/20

Atmega1284P INVADER Data Shield Specifications

1. Data Specifications

The data are made up with the logs of the ARTSAT1: INVADER satellite from 18:00:11UTC on February 28th, 2014 through 06:38:43UTC on August 31st, 2014.

a. 3518 records

b. Specification of the data: 28 items

	Item	Integral Figures	Decimal Figures	Form	Minimum Value	Maximum Value
1	Month	2	0	99	2	8
2	Day	2	0	99	1	31
3	Hour	2	0	99	0	23
4	Minute	2	0	99	0	59
5	Second	2	0	99	0	59
6	Solar cell current -Y2	3	1	999.9	0	250.9
7	Solar cell current +Y2	3	1	999.9	0	209.1
8	Solar cell current -Z	3	1	999.9	0	374.7
9	Solar cell current +Z	3	1	999.9	0	577.2
10	Solar cell current -Y1	3	1	999.9	0	193.0
11	Solar cell current +Y1	3	1	999.9	0	318.9
12	Solar cell current -X	3	1	999.9	0	418.3
13	Solar cell current +X	3	1	999.9	0	450.9
14	Battery temperature 1	2	0	s99	-8	28
15	Battery temperature 2	2	0	s99	-8	56
16	Battery temperature 3	2	0	s99	-7	64
17	Solar cell temperature -X	2	0	s99	-38	27
18	Solar cell temperature +Y1	2	0	s99	-38	27
19	Solar cell temperature +Y2	2	0	s99	-38	69
20	Solar cell temperature -Y1	2	0	s99	-38	41
21	Solar cell temperature +Z1	2	0	s99	-38	56
22	Solar cell temperature +Z2	2	0	s99	-38	56
23	Solar cell temperature -Z1	2	0	s99	-38	32
24	Solar cell temperature -Z2	2	0	s99	-38	32
25	Main OBC temperature	2	0	s99	-38	33
26	Gyro X	1	1	S9.9	-3.7	3.7
27	Gyro Y	1	1	S9.9	-3.7	3.7
28	Gyro Z	1	1	S9.9	-3.7	3.7

Note 1. The data file form is the ".csv" using ',' as delimiters.
2. The date and time are the UTC when the data were obtained.
3. The values are rounded down to the figures indicated above in each item, which are received from the "ARTSAT1: INVADER" satellite.
4. The '9' in the column "Form" means one numeral, as well as 's' means '-' (minus) in case the value is minus. The '-' is not included in the "Integral Figures".
5. The numbers of the figures are fixed, so the '0's are added in the upper.
6. The 'CR' code (0x0D) is added at the end of each record as the delimiter of records.

2. How to communicate with an "Arduino"

This shield is connected through the soft serial function on the Arduino.

You can select the 4th, 7th, 8th, or 12th digital connect line on the Arduino by jumper pins on this board.

The speed of the sending or receiving through the serial line is 9600 bps.

3. Data sending specifications

a. The data is started at the first, 18:00:11 UTC on February 28th, after entering the supply.

b. The timing to send data is selectable between every second or every 5 seconds. You can select the timing before you supply power to this board by using jumper pins connected 40th pin of Atmega1284P on this board. The timing is set as 1 second when you make open, 5 seconds when short.

c. The timing is changeable by the command from the Arduino.

d. When the record comes to the end, 06:38:43 UTC on August 31st, the next data starts from the first.

4. Start or stop sending data by the push switch on the board

a. Alternatively start or stop when you push the switch on the board

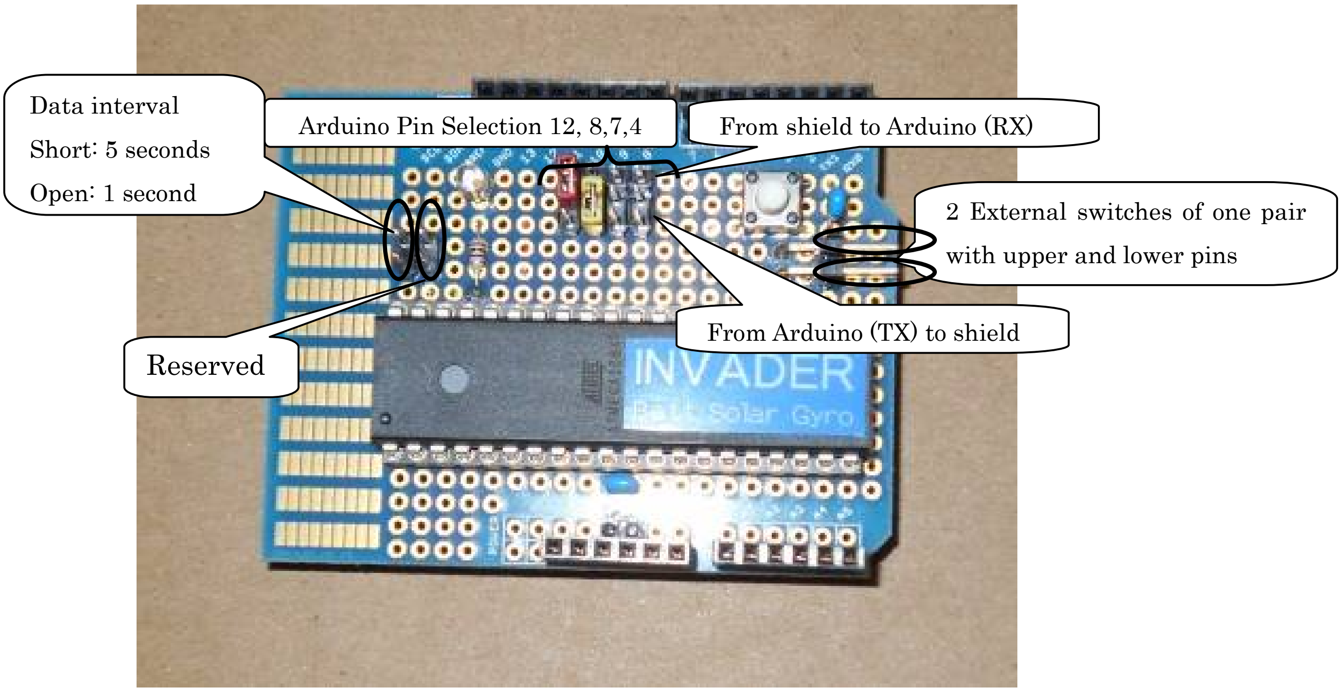
You can add external switches to pins on the board. You may need some additional circuits like an capacitor for a good function.

b. The LED on the board will light on while sending data, off while stopping. It is the same as sending or stopping by the command from the Arduino.

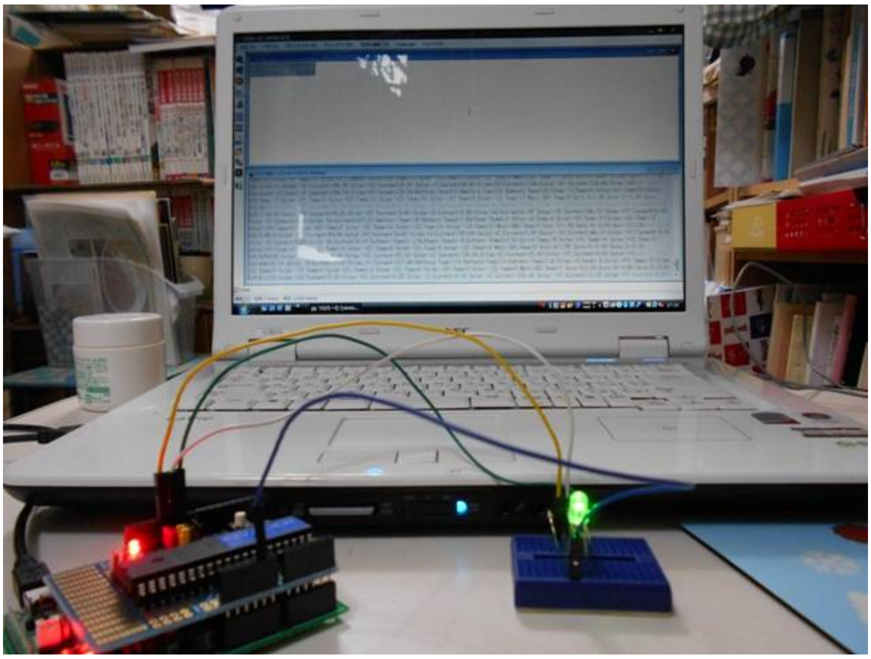
5. Commands through the serial line

Commands	Functions	Details	Remarks
B	Start sending data		Start sending data from the next of the former data or the set position with the set interval
S	Stop sending data	Suspend to send data	
K	Send one record		*Enable while stopping
1	Set 1 second as the interval		The interval will be changed after once sent a record with the former interval
5	Set 5 seconds as the interval		The interval will be changed after once sent a record with the former interval
R	Reset the data position at the first		*Enable while stopping
P	Set the data position	Set the data position as the record number following the 4 figures from 0000 to 3517	*Enable while stopping
M	Set the data position from March		*Enable while stopping
A	Set the data position from April		*Enable while stopping
Y	Set the data position from May		*Enable while stopping
J	Set the data position from June		*Enable while stopping
Y	Set the data position from July		*Enable while stopping
G	Set the data position from August		*Enable while stopping

6. Instruction of the jumper lines


Data interval
Short: 5 seconds
Open: 1 second
Arduino Pin Selection 12, 8,7,4
From shield to Arduino (RX)
2 External switches of one pair with upper and lower pins
From Arduino (TX) to shield
Reserved
INVADER

7. An example scene to use with the sample sketch



See the movie at https://www.dropbox.com/s/0oqcwssggzj4gnu/DSCN0602.wmv?dl=0

8. A sample sketch

```
/*
  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);

          }
}
```