Final Project Presentation

“Sensor data acquisition routines for AVR based embedded processors”

written by Shin Hyun Sup

2005. 8. 9
This project is to implement the program on the AVR128 that transmits GPS Information from PG31 (GPS Machine) to Computer.

AVR 128 Board transmits the GPS information to AVR 128 Board

AVR 128 Board receives the GPS information and extracts the data – Time, Latitude, Longitude, Altitude, Velocity

PG-31 transmits the GPS information to AVR 128 Board

AVR 128 Board translates the data and sends to Computer.
**Previous Research** - What have other done?

Studying embedded programming for AVR.
And developing, implementing the simple programs.

- **Implement the code for controlling LEDs and timer interrupt.**
  This program counts the numbers between 0 and 255 by LEDs. Then we assume that turning on is 1, turning off is 0. I learned the method of controlling port and using timer Interrupt by this program.

- **Implement the code for communication between AVR128 and computer by USART(RS- 232).**
  This program is to communicate between AVR128 Board and PC through USART. This program is used in Final Project to transmit Data. (PC, AVR128 Board, GPS)

- **Implement the code for accessing the EEPROM in the CPU.**
  The CPU of AVR128 Board has the EEPROM of 4096 bytes. I studied the method of reading data from EEPROM and writing to EEPROM.
Project - Software & Hardware Setting (1)

Software Setting
- Programmers Notepad 2 (WinAVR), HyperTerminal

Hardware Setting
- Super slim Notebook Computer, AVR128 Board, PG-31 (GPS Machine), Cables
Project - Software & Hardware Setting(2)

- Connect with PC by Hyper Terminal
- Load hex file to board from PC

This is USART0 to connect PC
This is USART1 to connect GPS
Prototype of Functions

// FUNCTIONS FOR GETTING GPS DATA

void getGPSData(void); // to Get GPS Data from AVR128 Board through
void ParseGPS(uint8_t buffer[80]); // to Get GPS Data of GGA that has the data of Time, Latitude, Longitude, Altitude
void ParseGPS_Velocity(uint8_t buffer[80]); // to Get GPS Data of RMC that has the data of Velocity
void printOutGPSData(void); // to print out GPS Data to PC through Hyper terminal
void initUSART(void); // to Initialize USART0, USART1

// FUNCTIONS FOR UART (IN UART_FUNCTION.C)

void USART_Init( int flag, unsigned int baud ); // to Initialize USART0, USART1
void USART_Transmit( int flag, unsigned char data ); // to Transmit character to USART0, USART1
void USART_TransmitString( char *str ); // to Transmit string to USART0
void USART_TransmitDigit( int Value ); // to Transmit digit to USART0
unsigned char USART_Receive( int flag ); // to Receive data from USART0, USART1
### Project - Result analysis(1)

<table>
<thead>
<tr>
<th>Type of data</th>
<th>GPS Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>$GP$ GSA</td>
<td>A,1,...........,50.0,50.0,50.0*05</td>
</tr>
<tr>
<td>$GP$ GSV</td>
<td>3,1,12,14,89,000,31,30,42,000,,05,41,000,00,22,35,000,00*7D</td>
</tr>
<tr>
<td>$GP$ GSV</td>
<td>3,2,12,25,34,000,00,11,15,000,00,09,12,000,00,18,04,000,00*78</td>
</tr>
<tr>
<td>$GP$ GSV</td>
<td>3,3,12,20,04,000,00,07,03,000,00,24,02,000,,06,-13,000,00*55</td>
</tr>
<tr>
<td>$GP$ RMC</td>
<td>174820.289,V,0000.0000,N,00000.0000,E,0.00,,290705,,*08</td>
</tr>
<tr>
<td>$GP$ GGA</td>
<td>174821.289,0000.0000,N,00000.0000,E,0.00,50.0,0.0,M,0.0,M,0.0,0000*7C</td>
</tr>
<tr>
<td>$GP$ GSA</td>
<td>A,1,...........,50.0,50.0,50.0*05</td>
</tr>
<tr>
<td>$GP$ RMC</td>
<td>174821.289,V,0000.0000,N,00000.0000,E,0.00,,290705,,*09</td>
</tr>
</tbody>
</table>
Final Project

Project - Result analysis(2)

Time : 003630.998, Latitude : 34 degrees, 4112 minutes, Longitude : -118 degrees, 26630 minutes, Altitude : 106.9, Velocity : 0.00
Time : 003635.998, Latitude : 34 degrees, 4112 minutes, Longitude : -118 degrees, 26630 minutes, Altitude : 105.9, Velocity : 0.00
Time : 003640.998, Latitude : 34 degrees, 4113 minutes, Longitude : -118 degrees, 26630 minutes, Altitude : 105.3, Velocity : 0.00
Time : 003645.998, Latitude : 34 degrees, 4113 minutes, Longitude : -118 degrees, 26630 minutes, Altitude : 106.9, Velocity : 0.00
Time : 003650.997, Latitude : 34 degrees, 4112 minutes, Longitude : -118 degrees, 26630 minutes, Altitude : 107.5, Velocity : 0.00
Time : 003655.997, Latitude : 34 degrees, 4112 minutes, Longitude : -118 degrees, 26630 minutes, Altitude : 108.7, Velocity : 0.00
Time : 003700.997, Latitude : 34 degrees, 4112 minutes, Longitude : -118 degrees, 26630 minutes, Altitude : 106.9, Velocity : 0.00
Time : 003705.996, Latitude : 34 degrees, 4112 minutes, Longitude : -118 degrees, 26631 minutes, Altitude : 105.9, Velocity : 0.00

hhmmss : this type is UTC (Universal Time)

Positive value means “North”
Negative value means “South”
Positive value means “East”
Negative value means “West”
I couldn’t analyze the data as soon as AVR128 Board received, because the data from GPS is transmitted continually.

So I made enough space to receive the data from GPS and saved the data in the buffer.

And then, I analyzed the data by the special characters of ‘$', ‘,’ ,'*'.

The method of Embedded-based programming is nearly same to the method of general PC based programming.

But there are a little differences in the part of variable and memory accessing, so I felt difficult in using them.
Conclusion

This project is my first project that is related with Embedded System. But the environment of AVR Development is based on C Language. So it’s not too difficult.

At fist time, I had some mistakes, because I was not used to use manual of hardware.

But whenever I felt difficulty, Harish helped me.

I appreciate him. Thank you!