Report
The AVR interface to GPS and EEPROM

1. The AVR interface to GPS

Used USART (The Universal Synchonous and Asynchronous serial Receiver and Transmitter) interrupt for interface betweem AVR and GPS, the USART transmit and receive serial data by TxD and RxD respective it use in Asynchronous mode transfer data with baut rate is 9600bps and has frame format 1 start bit, 8bits data and 1 stop bit. For the GPS, it operate in Full-time operation mode for transfer data 124bytes per second and setup the periodical output data, there are Position data, Date/Time data, GPS satellite information and Error index information.

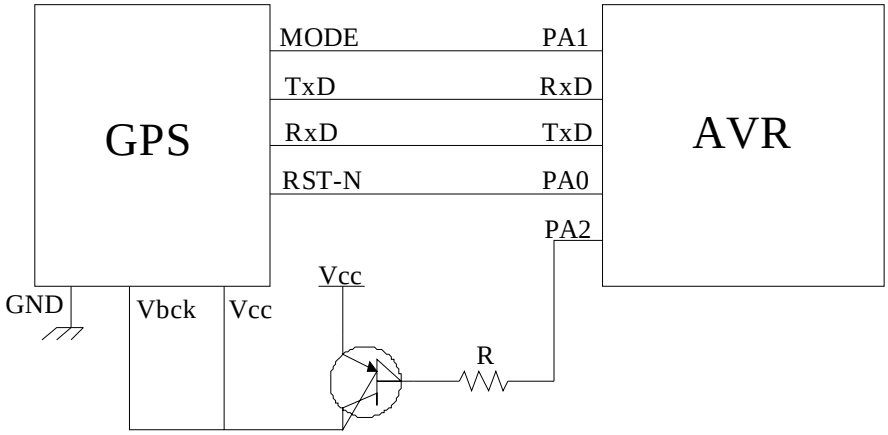


Figure 1 show the AVR interface to GPS

1.1 The control process of AVR to GPS

As show in flow chart 1

. First set power-on for GPS by clear bit PA2
. After this 300ms set bit PA0 for reset GPS
. Send control command GPS reciver parameters such as:
- Setup Full-time operation mode (10000001)
- Setup Periodical Output Data (10001111)

. Wait for interrupt receiver complete
. Used USART interrupt receive data from GPS then move data into SRAM by assembly program
. SRAM address has 400 spaces begin from 0x0060 to 0x045F if write data into SRAM is full (0x045F) the assemby program will return to write next data into 0x0060 until 0x045F then repeat again.

2. the AVR interface to EEPROM

Used SPI (Serial Peripheral Interface) interface for transfer data between AVR and EEPROM the AVR will read data from SRAM 124 bytes and write into EEPROM, before write that data into EEPROM the AVR must write 2 bytes sychronous patter and 2 bytes CRC patter after write data finish. So 1 frame data will be 128 bytes and repeat this process for next 1 frame. For the EEPROM, we use 512Kbyte when the AVR write data into EEPROM until it is full the assembly program must finish and also stop system.

The SPI consist of two operation mode: writing mode and reading mode by 4 pin as bellow:

- CS (Chip Select) is active low use to enable or disable the EEPROM
- MOSI (Master out - Slave in) use for send data into EEPROM
- MISO (Master in - Slave out) use for read data from EEPROM
- SCK (Clock Signal) is generate clock signal

. Writing mode

- Set CS active low to enable EEPROM
- Send instruction code 0x02 by MOSI line
- Send Address code 16bits by MOSI line
- Send data 8bits by MOSI line
- Set CS active high to disable EEPROM

. Reading mode

- Set CS active low to enable EEPROM
- Send instruction code 0x03 by MOSI line
- Send Address code 16bits by MOSI line
- Read data 8bits by MISO line
- Set CS active high to disable EEPROM

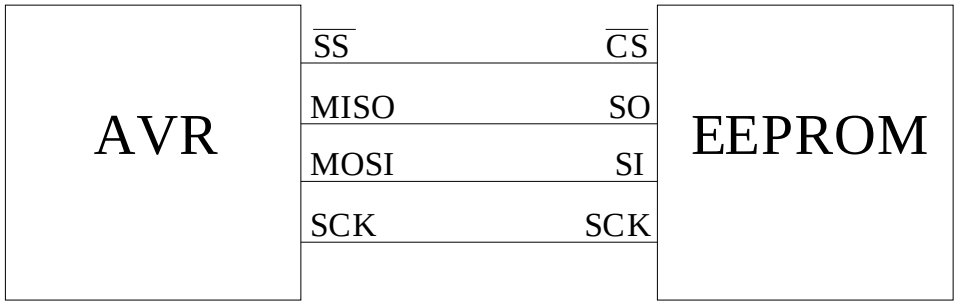


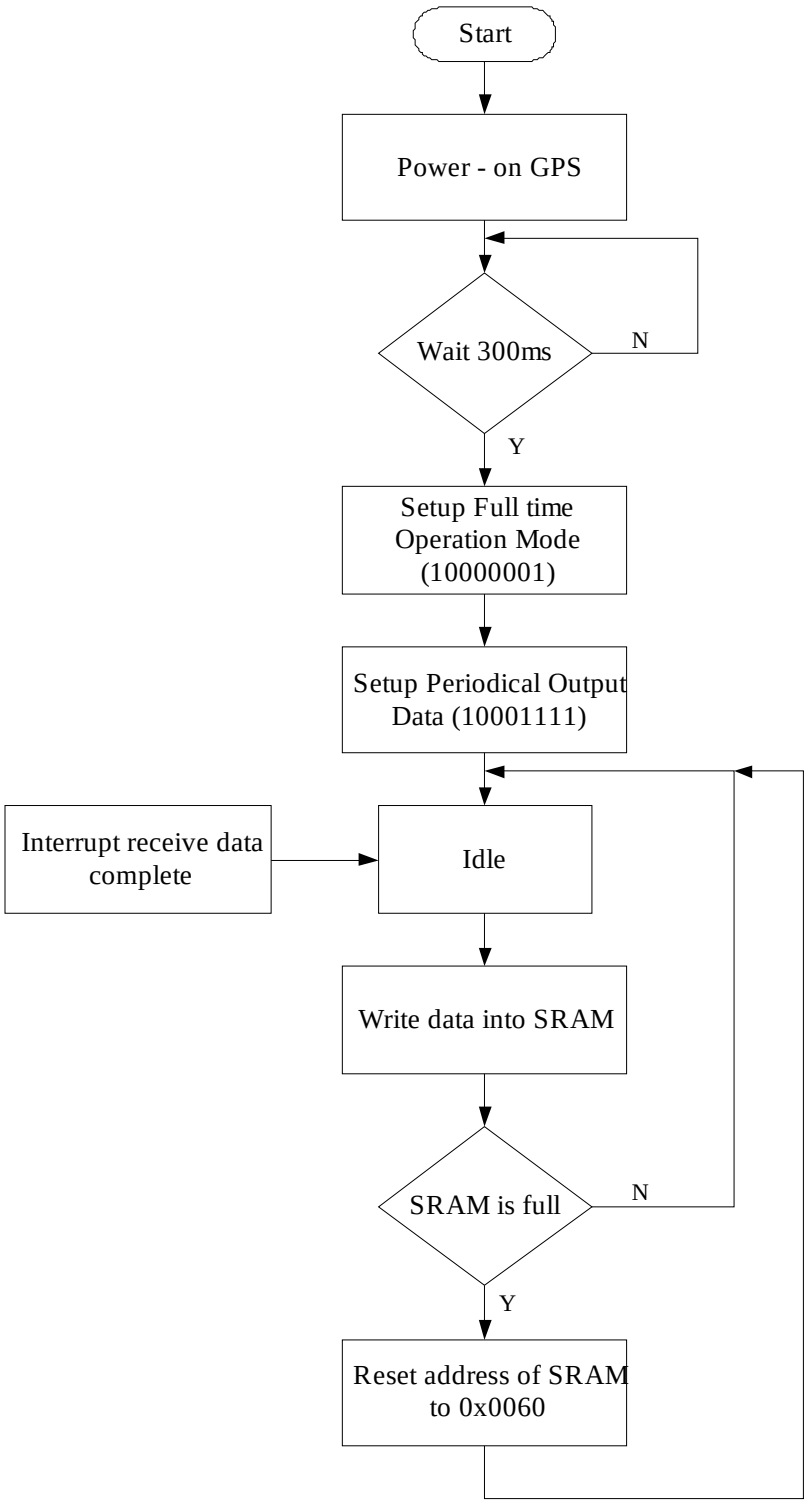
Figure 2 show the AVR interface to EEPROM

2.1 The control process of AVR to EEPROM

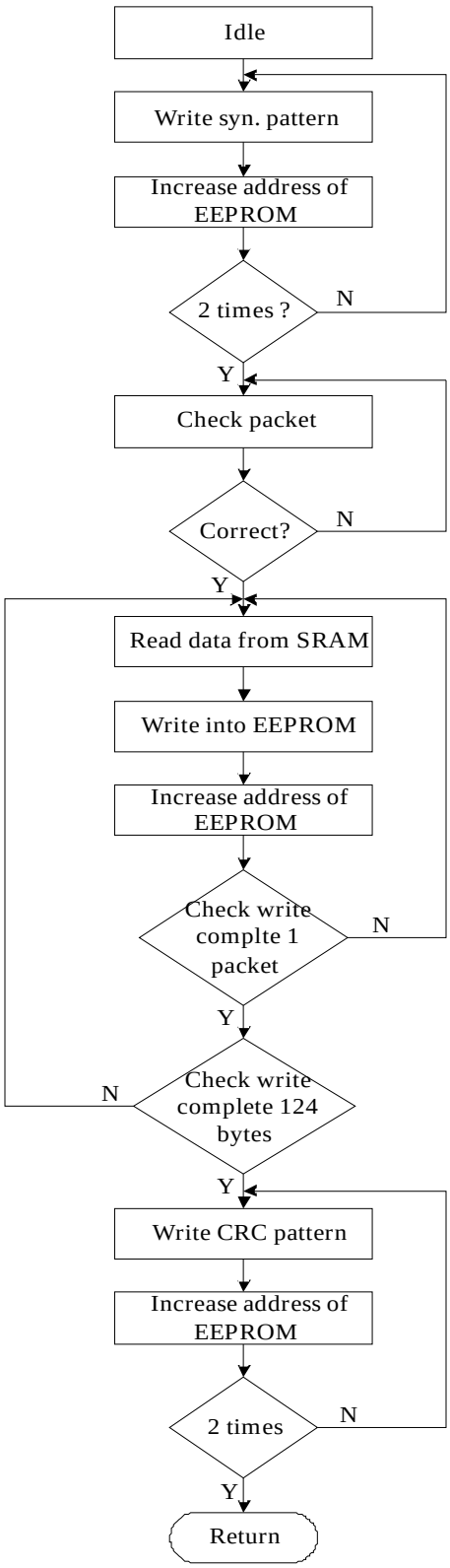
As show in flow chart 2

. Initial value for SPI interface
. Write 2bytes synchronous pattern into EEPROM
. Read data from SRAM
. Check sum one packet (it is correct or not)
. If that data correct then write into EEPROM until equal to 124bytes, if not correct the AVR go to read again until correct when it still not correct the AVR will send message to tell system that is error.
.Write 2 bytes CRC pattern into EEPROM (this is finish 1 frame data 128 bytes)
.Increase address of EEPROM then continious write 1 frame data
.Check EEPROM's address it is full or not, if it is not full the AVR continious write data if it is full the system will stop at that time.

Flow chart 1: Setup and Receive data from GPS



Flow chart 2: Write data into EEPROM (1frame = 128 bytes)


Idle
Write syn. pattern
Increase address of EEPROM
2 times ?
N
Y
Check packet
Correct?
N
Y
Read data from SRAM
Write into EEPROM
Increase address of EEPROM
Check write complte 1 packet
N
Y
Check write complete 124 bytes
N
Y
Write CRC pattern
Increase address of EEPROM
2 times
N
Y
Return