Introduction to Electric Vehicles

Monitor Program

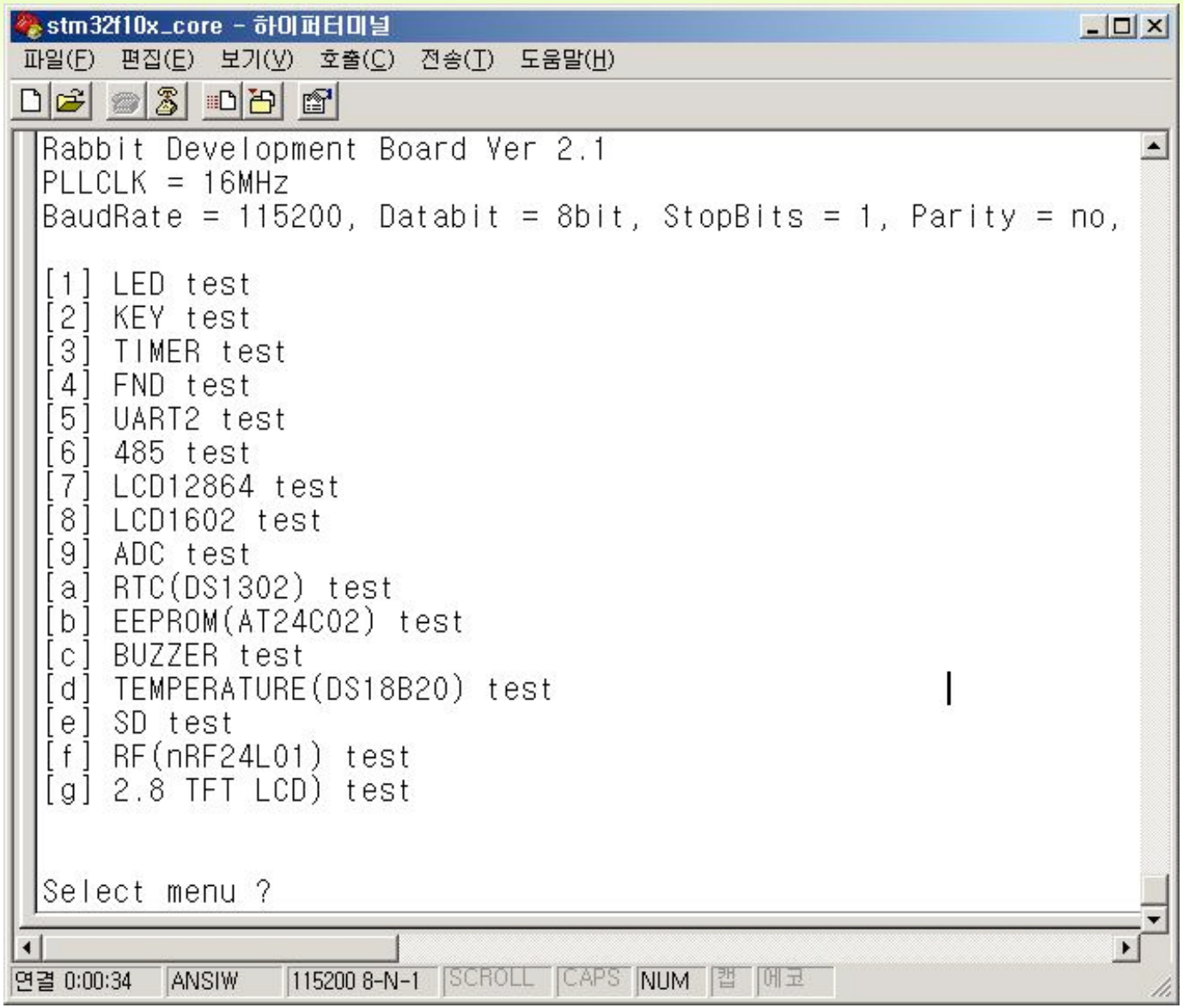
Joonki Hong
Dept. of Electrical Engineering
Korea Advanced Institute of Science and Technology
joonki@cad4x.kaist.ac.kr

KAIST

Goals

- Implement monitor program
- Functions have to be implemented
  - UART communication (by polling)
    - putChar(), putString(); //Already implemented
    - unsigned char genChar();
    - char* getString();
    - myPrintf(const char *format,…)
      - va_start();
      - va_end();
  - Memory modification
    - Read one byte from typed memory address
    - Read N bytes from typed memory address
    - Write one byte from typed memory address and one byte data
    - Write N bytes from typed memory address and N bytes data
  - Super loop approach
    - Handle user input
    - Support menu functions

Monitor program example



Memories

- Read the data sheet pp. 18 - 36

Memory		Mnemonic	AT90CAN32	AT90CAN64	AT90CAN128
Flash	*Size*	Flash size	32 K bytes	64 K bytes	128 K bytes
	Start Address	-	0x00000		
	End Address	Flash end	0x07FFF(1) 0x3FFF(2)	0x0FFFF(1) 0x7FFF(2)	0x1FFFF(1) 0xFFFF(2)
32 Registers	*Size*	-	32 bytes		
	Start Address	-	0x0000		
	End Address	-	0x001F		
I/O Registers	*Size*	-	64 bytes		
	Start Address	-	0x0020		
	End Address	-	0x005F		
Ext I/O Registers	*Size*	-	160 bytes		
	Start Address	-	0x0060		
	End Address	-	0x00FF		
Internal SRAM	*Size*	ISRAM size	2 K bytes	4 K bytes	4 K bytes
	Start Address	ISRAM start	0x0100		
	End Address	ISRAM end	0x08FF	0x10FF	0x10FF
External Memory	*Size*	XMem size	0-64 K bytes		
	Start Address	XMem start	0x0900	0x1100	0x1100
	End Address	XMem end	0xFFFF		
EEPROM	*Size*	E2 size	1 K bytes	2 K bytes	4 K bytes
	Start Address	-	0x0000		
	End Address	E2 end	0x03FF	0x07FF	0x0FFF

Memories

- Flash program memory
  - Write/erase the firmware to flash memory
  - Flash memory space is divided into two sections
    - Boot program section
    - Application program section
- SRAM data memory
  - Volatile memory
  - Fast operation
- EEPROM data memory
  - Non-volatile
  - Slow write/read operation

Homework (project) #3

- Printed report is required
  - Until May 22nd, Friday class. (one week)
  - No delayed paper accepted
- Contents should include
  - Source code
  - Detailed explanation about the source code
  - Screenshot of the operation

Functions for undetermined number of input

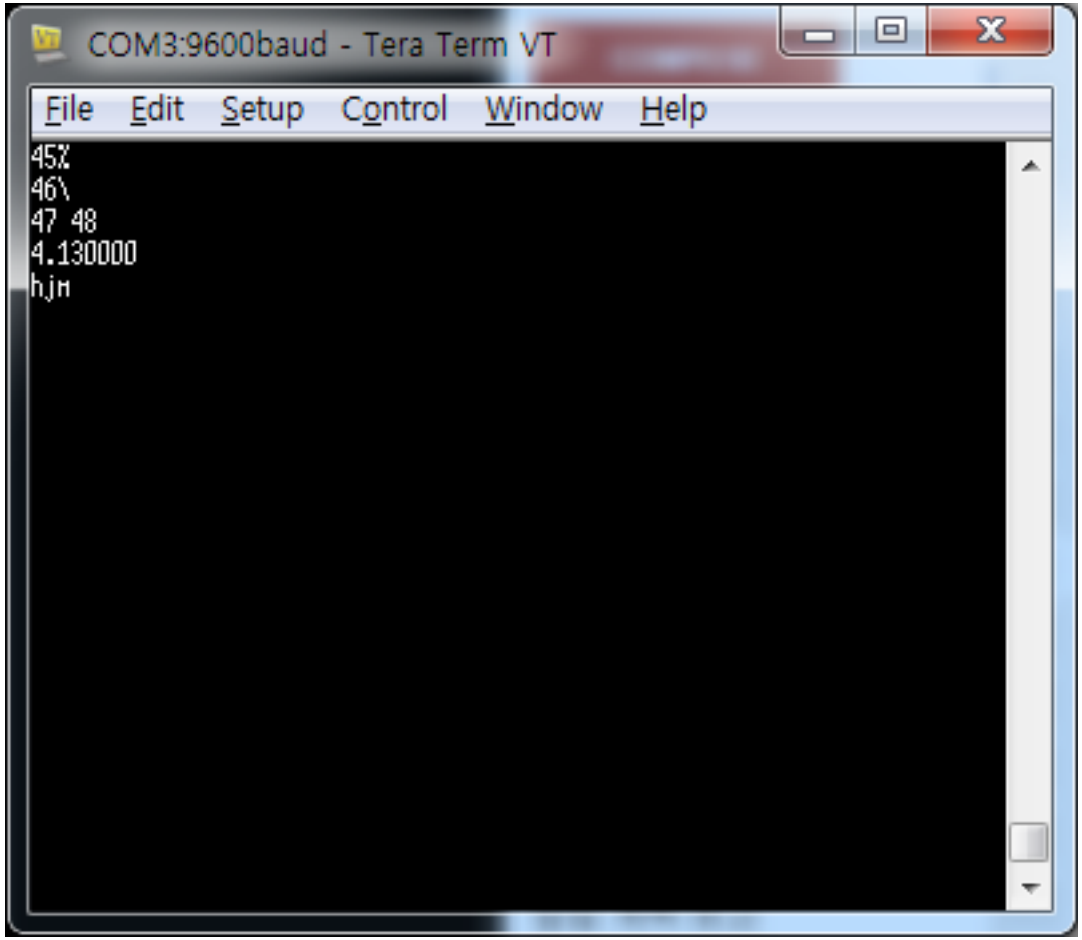
- #include <stdarg.h>
  - va_list()
  - va_start()
  - va_arg()
- #include<stdlib.h>
  - itoa()
  - atoi()
  - dtostrf()
- #include<string.h>
  - strtok()
- #include”avr/eeprom.h”
  - eeprom_read_byte()
  - eeprom_write_byte()

Result_Joonki (1)

- Result 1
  - myPrintf

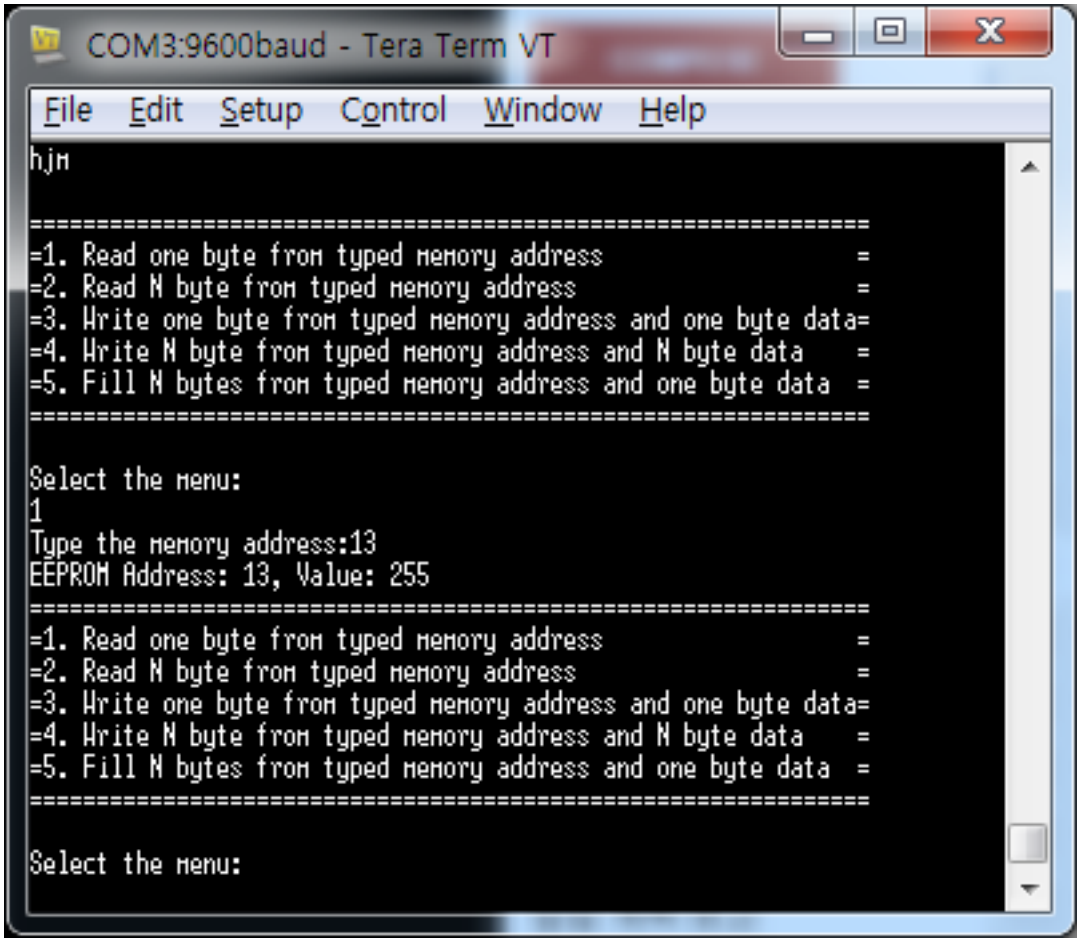
```
myPrintf("%d%%\n\r",45);
myPrintf("%d\\\n\r",46);
myPrintf("%d %d\n\r",47,48);
myPrintf("%f\n\r",4.13);

myPrintf("%c",c);
myPrintf("%s\n\r\n","jm");
c = 'h';
putString(getString());
```



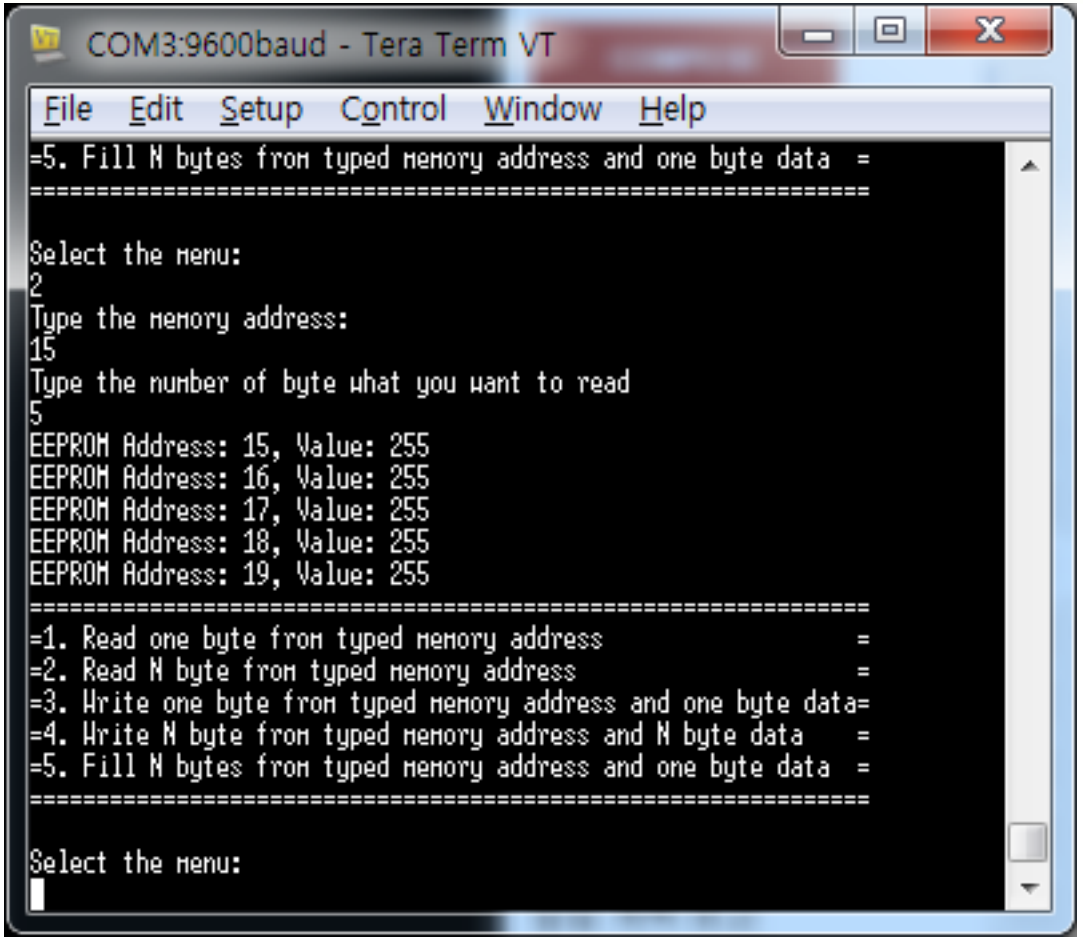
Result_Joonki (2)

- Function 1
  - Read the data from the data address (13)



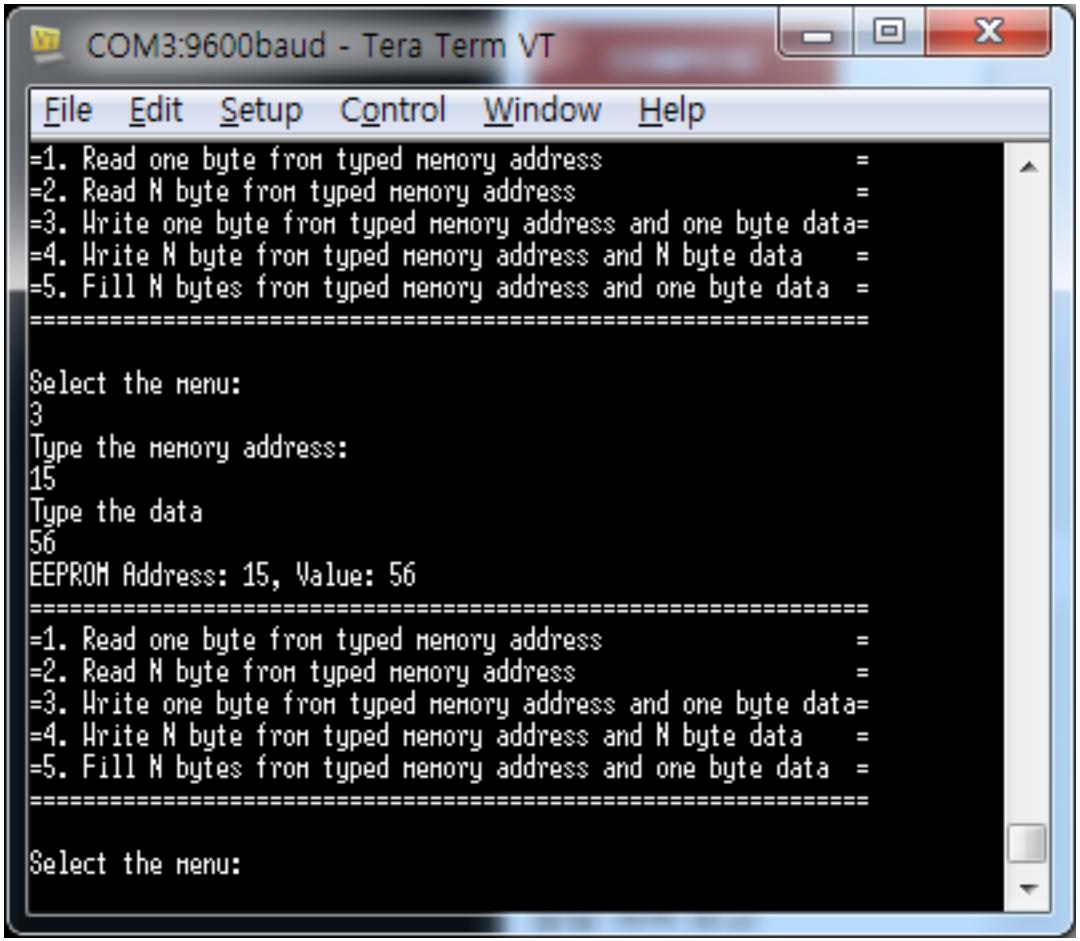
Result_Joonki (3)

- Function 2
  - Read the data from the data address (15 ~ 19)



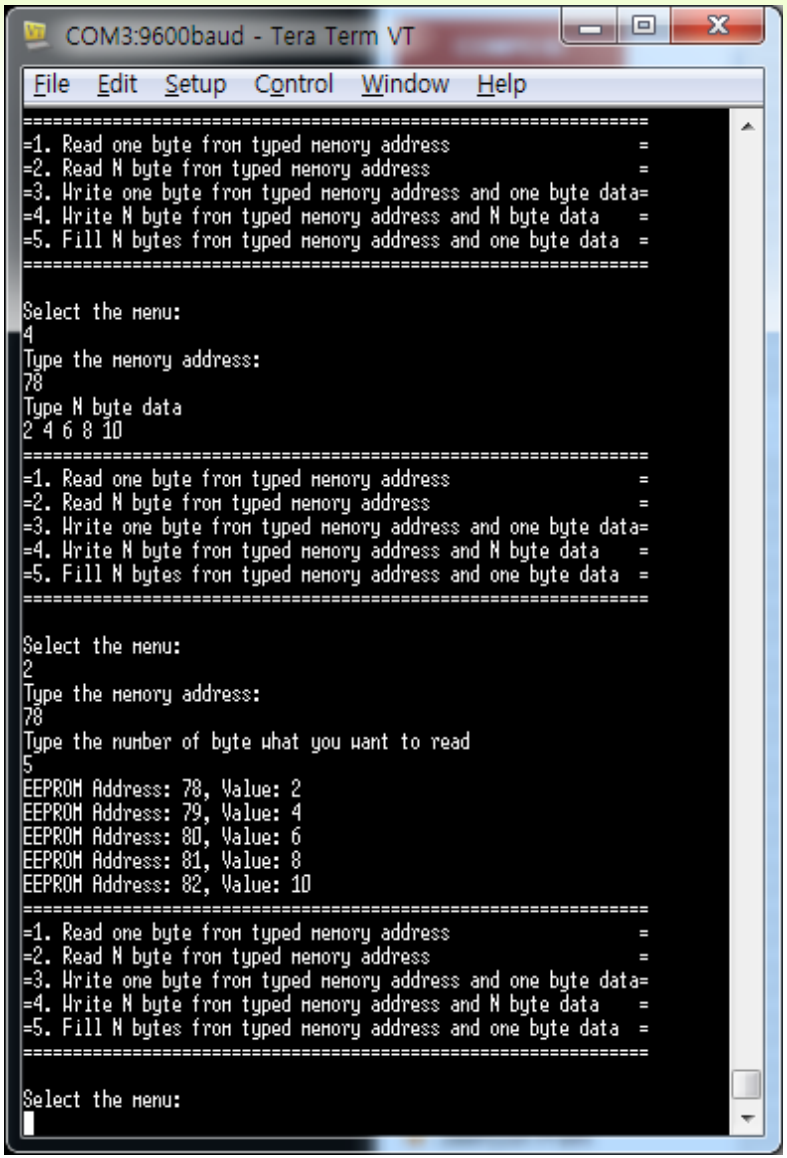
Result_Joonki (4)

- Function 3
  - Write the value ’56’ to the data address (15)



Result_Joonki (5)

- Function 4
  - Write 2, 4, 6, 8, 10 to the data address (78 ~ 82) and read the data address



Result_Joonki (6)

- Function 5
  - Write the value ’99’ to the data address 45 ~ 54