Singapore Polytechnic
School of Electrical and Electronics Engineering
ET0104 Embedded Computer Systems DECC 3FT/4EO

Tutorial 12 Embedded C and Operating systems

1. Portability, Efficient, Productivity (any other reasonable, is OK)

   For embedded C:
   * Program code is in EPROM which is normally tens of Kbytes
   * Data and stack area are in RAM which is only hundreds of bytes
   * Standard I/O devices like graphics display, keyboard, hard disk are not present, so are the functions associated with these devices, like `printf, scanf.`
   * We can also build a header files to define all the special registers, control or flag bits, then can directly use their standard names as in assembly language.

   Those functions which require high processing speed may need to be programmed in assembler.

2. Save on RAM
   * Use "unsigned char" variables - reduce the usage of RAM, and shorten the generated code length.
   * Use bit flags instead of bytes for true/false variables: - less RAM.
   * Use ROM for storage of constants, tables, messages and other non-volatile data: - less RAM.
   * Try not to pass data in functions, that will take up more space on stack.
   * Use global variables instead of local variables to save RAM space and increase speed.

   Speed
   * Use pointer for array elements will be more efficient in terms of speed and code size.

3. ```
   data = _inp(0x132);
   _outp(data, 0x134);
   ```
   ```