Overview

- Parallel Ports
  - Data Direction Register
- Sample Codes
Parallel Ports

- Bytewise Data transfer
  - 8-bit transfer (read/write) in one step

- 10 Parallel Ports
  - A, B, E, H, J, K, M, P, S, T

Reference: datasheet of mc9s12dg128 (can be found at courses website)
Each port has a controlling register which is called Data Direction Register (DDR)

- Ports are bi-directional and bits of a port can be configured for input or output (but not the same time!)
- Direction is determined by specific bit/s in the corresponding port DDR (0=input, 1=output)

**Important Notes**
- Out-of-reset all ports are input!
- While using an external output source do not directly use +5V. You should use a resistor (2.2 or 4.7 kΩ) to prevent unintentional burn of port (:}
Sample Code #1

- Assume you use PortA for output and PortB for input

LDAA #$FF;                  \(\text{Configure bits of DDRA}\)
STAA DDRA;                  \(\text{Write to DDRA (Reserved Address: }$0002)\)
STAB PORTA;                 \(\text{Write to PortA (Reserved Address: }$0000)\)
LDAA #$00;                  \(\text{Configure bits of DDRB}\)
STAA DDRB;                  \(\text{Write to DDRB (Reserved Address: }$0003)\)
LDAB PORTB;                 \(\text{Read from PortB (Reserved Address: }$0001)\)

Alternatively (using C)

DDRA = 0xFF;                \(\text{Configure bits of DDRA}\)
PORTA = my_char;            \(\text{Write to PortA}\)
DDRB = 0x00;                \(\text{Configure bits of DDRB}\)
my_char = PORTB;            \(\text{Read from PortB}\)

Important Note:
- PORTs and their DDRs have reserved physical memory addresses! (See reference datasheet of mc9s12dg128 for more information)
#include <hidef.h>  /* common defines and macros */
#include <mc9s12dg128.h>  /* derivative information */
#pragma LINK_INFO DERIVATIVE "mc9s12dg128b"

void main(void) {

    unsigned char my_num[] = "0123456789ABCDEF"; // Initialize character array
    unsigned int i,j;

    DDRA= 0xFF; // Make all bits of Port A output

    for(i=0; i<16; i++){
        PORTA = my_num[i]; // Display the ith character of array
        for(j=0; j<60000; j++)
        }

    while(1); // Stay here forever
}