

Laboratory 02: Bitwise Logical Operations in C

Prof. Ken Short

2/5/2022

© Copyright Kenneth Short 2022

1

Overview

- The purpose of this laboratory is to provide you with experience in configuring port pins and writing simple programs that require bit manipulation using C's bitwise logical and shift operators.

2/5/2022

© Copyright Kenneth Short 2022

2

Laboratory 02 Laboratory Tasks

- ❑ Task 1: A Simple 8-bit Parallel Input Port and a Simple 8-bit Parallel Output Port
- ❑ Task 2: Software Read-Modify-Write
- ❑ Task 3: A Simple Combinational Function Using Bitwise Logic Operators and Shifting
- ❑ Task 4: A Simple Combinational Function Using Named Bits

2/5/2022

© Copyright Kenneth Short 2022

3

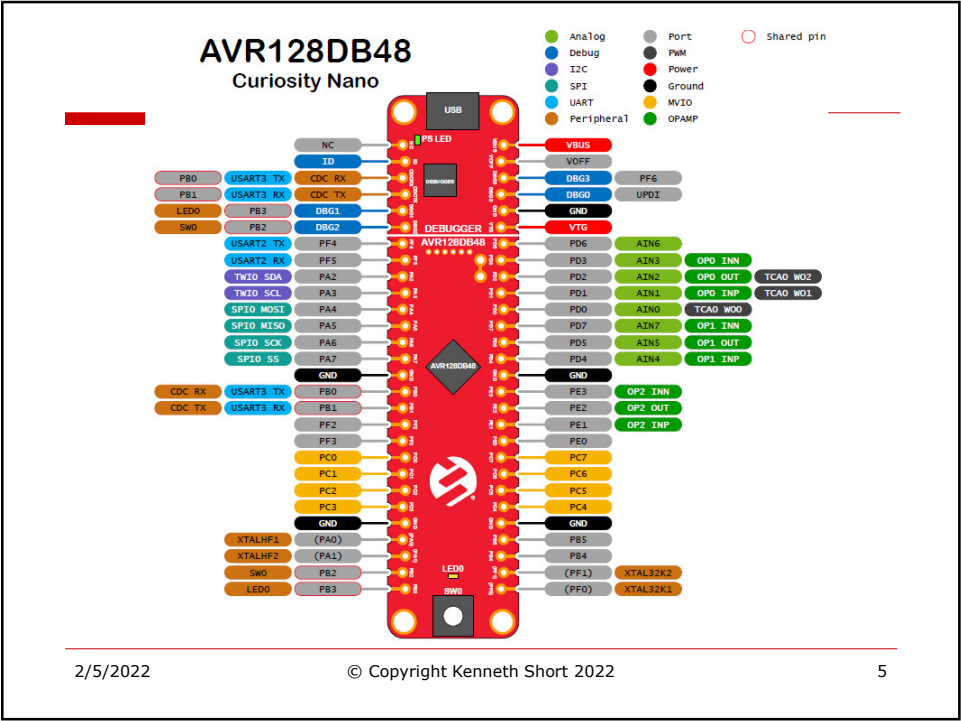
Task 1: A Simple 8-bit Parallel Input Port and a Simple 8-bit Parallel Output Port

- ❑ An 8-bit parallel input port and an 8-bit parallel output port are to be implemented.
- ❑ The input port is connected to one side of an 8 position SPST DIP switch. The other end of each switch is connected to ground. The input pins for this port must have their internal pull-up resistors enabled in software. When a switch is open it must read as 1 and when it is closed it will be read as a 0.
- ❑ The output port drives 8 LEDs of a 10 element LED bargraph. The top two elements of the bargraph are not used. The LEDs cathodes are connected to the pins of the output port. Use a 330 ohm SIP resistor for current limiting.
- ❑ The system continually reads the switches and turns ON the LEDs corresponding to the switches that are open.
- ❑ The input port will consist of pins: PA7, PA6, PA5, PA4, PA3, PA2, PC1, and PC0. For the output port use PD7 through PD0.

2/5/2022

© Copyright Kenneth Short 2022

4



Task 2: Software Read-Modify-Write

- ❑ You must write a program named `read_modify_write_sftw_sw0` to accomplish a read-modify-write operation of the output of `VPORTD_OUT`.
- ❑ When this program first starts execution, it reads the input port of Task 1 and uses that value to control the LEDs of `VPORTD_OUT`, just as in Task 1.
- ❑ After that, whenever `SW0` is pressed, the program uses the values of bits 3 down to 0 of the input port to control LEDs 6 down to 3 of the output port. The other LEDs of the output port must not have their values changed.

Task 3: A Simple Combinational Function Using Bitwise Logic Operators and Shifting

- ❑ The C bitwise logical operators that you have studied provide all the logical operations needed to carry out Boolean function computations.
- ❑ The function that you must implement for this task is basically a 3-input OR gate or odd parity checker. Use a SOP expression.

C	B	A	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Task 4: A Simple Combinational Function Using Named Bits

- ❑ In Lecture 4 we discussed how to use a union of an `uint8_t` and a structure with bit fields to give names to each bit in a byte of data. Using this approach it is relatively easy to write (and compute) a sum-of-products expression in C. You are to use this approach to implement the function described in Task 3.
- ❑ You can use the header file `data.h` from that lecture or modify it to have more descriptive names for this particular function.
- ❑ Write a program named `xor3_named_bits` that implements this function using named bits via the bitfield approach.