GPIO Ports

Reference:

1. Reference System Design
2. Adafruit Motor Shield - Part 1

You may assume the PUD in MCUCR is cleared (normal operating mode).

DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

Consider the following C code example from Section 13.2.4 Reading the Pin Value from the datasheet.

Code Example 1

```
unsigned char i;
...
/* Define pull-ups and set outputs high */
/* Define directions for port pins */
PORTB = (1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0);
DDRB = (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0);
/* Insert nop for synchronization*/
__no_operation();
/* Read port pins */
i = PINB;
...
```

1. What hexadecimal value would be placed in PORTB and DDRB?

PORTB = 0xC3 DDRB = 0x0F

2. Which pins are defined as inputs with pull-up resistors

PINB7 and PINB6

3. Which pins are defined as outputs, initialized to logic 1

PINB1 and PINB0

Consider the following C code example from the Adafruit Part 1 SPI pdf document.

Code Example 2

```
void AFMotorController::latch_tx(void) {
 uint8_t i;

 //LATCH_PORT &= ~_BV(LATCH);
 digitalWrite(MOTORLATCH, LOW);     // - Output register clock low

 //SER_PORT &= ~_BV(SER);
 digitalWrite(MOTORDATA, LOW);      // - Serial data bit = 0

 for (i=0; i<8; i++) {              // - Shift out 8-bits
    //CLK_PORT &= ~_BV(CLK);
    digitalWrite(MOTORCLK, LOW);    //    - Shift clock low

    if (latch_state & _BV(7-i)) {   //    - Is current bit of
     //SER_PORT |= _BV(SER);                latch_state == 1
     digitalWrite(MOTORDATA, HIGH); //    -   Yes, serial data bit = 1
    } else {
     //SER_PORT &= ~_BV(SER);
     digitalWrite(MOTORDATA, LOW);  //    -   No, serial data bit = 0
    }
    //CLK_PORT |= _BV(CLK);
    digitalWrite(MOTORCLK, HIGH);   //    - Shift clock high, rising edge
 }                                  //      shift bit into shift register
 //LATCH_PORT |= _BV(LATCH);
 digitalWrite(MOTORLATCH, HIGH);    // - Output register clock high,
rising
}
```

4. In the first code example, variable **i** is defined as an unsigned character. Variable i is impicitly assumed to be of length 8-bits. How would you explicitly declaring i as a data type unsigned integer of length 8 bits?

```
unint8_t i;
```

In the Adafruit Part 1 SPI pdf document, _BV(i) is defined as a macro which evaluates to a byte having only the *i* th bit set. It's defined in avr/str_defs.h as:

```
#define _BV(bit) (1 << (bit))
```

5. Rewrite the PORTB assignment statement from the first code example using the _BV(i) macro.

```
PORTB = _BV(PB7) | _BV(PB6) | _BV(PB1) | _BV(PB0);
```

6. In the last problem bits 5, 4, 3, and 2 were cleared. Rewrite the answer to the previous question so these bits are not modified.

```
PORTB |= _BV(PB7) | _BV(PB6) | _BV(PB1) | _BV(PB0);
```

7. How would you configure the PORT B pin 1 as an output, initialized to logic 0 without modifying any other bits in the DDRB and PORTB registers?

```
DDRB |= _BV(DDB1);
```

```
PORTB &= ~_BV(PORTB1);
```

8. What Arduino Digital pin is mapped to PORT B pin 1? (Hint: look at Figure 4 in the "Adafruit Part 1 SPI" pdf document)

```
Digital Pin 9
```

9. How would you configure the PORT B pin 1 as an output, initialized to logic 0 using Arduino instructions? (Hint: Have you read this Arduino Tutorial)

```
pinMode(9, OUTPUT);
digitalWrite(9, LOW);
```

10. How would you configure the PORT B pin 4 as an input, with a pull-up resistor using Arduino instructions?

```
pinMode(12, INPUT);
digitalWrite(12, HIGH);
```