

Willis T. Allstead
 Professor Egbert
 CPE 301
 5 October 2016

HW 4

2)

a)

Port			
PC6	RXD		
PD1	TXD		
PD2	INT0		
PD3	INT1	PWM	OC2B
PD4	XCK	T0	
PB6	XTAL1	OSC1	
PB7	XTAL2	OSC2	
PD5	T1	PWM	OCOB
PD6	AIN0	PWM	OC0A
PD7	AIN1		
PB0	CLK0	ICP1	
PB1	OC1A	PWM	
PB2	OC1B	PWM	SS
PB3	OC2A	PWM	MOSI
PB4	MISO		
PB5	SCK		
PC4	SDA		
PC5	SCL		

b)

Port				
PB1	PCINT5	SCK		
PB2	PCINT2	MOSI		
PB3	PCINT3	MISO		
PB4	OC2A	PCINT4	PWM	
PB5	OC1A	PCINT5	PWM	
PB6	OC1B	PCINT6	PWM	
PB7	OC0A	PCINT7	PWM	OC1C
PD0	INT0	SCL		
PD1	INT1	SOA		
PD2	INT2	RXD1		
PD4	INT3	TXD1		

3)

6.2)

```

unsigned char *portDDRB = (unsigned char*) 0x24;
unsigned char *portDataB = (unsigned char*) 0x25;
unsigned char *portPinB = (unsigned char*) 0x23;
char test[16] =
{'A','B','C','D','E','F','0','1','2','3','4','5','6','7','8','9'};
unsigned int index = 0;

```

```

void changeDisplay(char);

```

```

void setup() {
    *portDDRB |= 0xFF; // configure all bits as outputs
}

```

```

void loop() {
    delay(2000);
    if (index == 16) { // restart loop
        index = 0;
    }

```

```

    changeDisplay(test[index]);

```

```

    index++;
}

```

```

void changeDisplay(char character) { // A->F & 0->9
    switch(character) {
        case 'A':
            portDataB |= 0x77; // 1110111
            break;
        case 'B':
            portDataB |= 0x1F; // 0011111
            break;
        case 'C':
            portDataB |= 0x4E; // 1001110
            break;
        case 'D':
            portDataB |= 0x3D; // 0111101
            break;
        case 'E':
            portDataB |= 0x4F; // 1001111
            break;
        case 'F':
            portDataB |= 0x47; // 1000111
            break;
        case '0':

```

```

        portDataB |= 0x7E; // 1111110
        break;
    case '1':
        portDataB |= 0x30; // 0110000
        break;
    case '2':
        portDataB |= 0x6D; // 1101101
        break;
    case '3':
        portDataB |= 0x79; // 1111001
        break;
    case '4':
        portDataB |= 0x33; // 0110011
        break;
    case '5':
        portDataB |= 0x5B; // 1011011
        break;
    case '6':
        portDataB |= 0x5F; // 1011111
        break;
    case '7':
        portDataB |= 0x70; // 1110000
        break;
    case '8':
        portDataB |= 0x7F; // 1111111
        break;
    case '9':
        portDataB |= 0x73; // 1110011
        break;
    default:
        portDataB |= 0x01; // 0000001;
    }
}

```

6.3)

```
unsigned char inputPin = *input;
```

```
// call the debounce function and supply the initial state of the inputPin.
```

```
bool debounce(unsigned char initialState) {
    delay(20);
    if (*input != initialState) {
        return true; // valid transition
    } else {
        return false; // invalid transition
    }
}

```

6.4)

```

unsigned char *portDDRB = (unsigned char*) 0x24;
unsigned char *portDataB = (unsigned char*) 0x25;
unsigned char *portPinB = (unsigned char*) 0x23;

bool debounce(unsigned char);

void setup() {
    *portDDRB |= 0x01; // configure bit 0 as an input from push-button
    *portDataB |= 0x01; // turn on bit 0 pull up;
}

void loop() {
    unsigned char previousSample;
    unsigned char currentSample;
    currentSample = *portPinB & 0x01;
    if (currentSample != previousSample) { // change occurred, check with
debounce
        if (debounce(previousSample)) { // transition was valid
            Serial.print("This input changed and was deemed a valid
transition.")
        }
    }
}

bool debounce(unsigned char initialState) {
    delay(20);
    unsigned char currentInput = *portPinB & 0x01;
    if (currentInput != initialState) {
        return true; // valid transition
    } else {
        return false; // invalid transition
    }
}

```