Fall 2020

ECET 209

Assessment #R

Digital Inputs and Outputs

|  |  |
| --- | --- |
| **Student Name** | **Student ID** |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S** | **Topic** | **Maximum Points** | **Earned Points** | **Feedback**  **Comment(s)** |
| **1** | **C code and simulation for Digital Output** | **25** |  |  |
| **2** | **C code and simulation for Digital Output** | **25** |  |  |
| **3** | **C Code: Digital Input and Output** | **25** |  |  |
| **4** | **C Code: Digital Input and Output** | **25** |  |  |
| **5** | **Bonus** | **5** |  |  |
|  | **Total** | **100** |  |  |

NOTES:

• Hand written reports will not be accepted (except if requested by Instructor)

• The assessment is out of 100 marks

• Up to 20 marks off for untidy / incompliant reports

1. **C code and simulation for Digital Output (25 Points)**
2. **Analyze the following C Language program**

**Explain in detail each code line using comments (15 p)**

|  |
| --- |
| void main()  {  /\*\*  TRISB is used to set the port direction  PORTB is used to read/write the port pin values  \*\*/  TRISB=0x00; // Sets PORTB as the output  PORTB=0x00; // Toggle PORTB off  TRISD=0x00; // Sets PORTB as the output  PORTD=0x00; // Toggle PORTD off  while(1)  {  PORTB=0xF0; // State 1 – lower 4 bits in PORTB for output  PORTD=0X0F; // State 1 - – lower 4 bits in PORTD for output  Delay\_ms(2000);// Delay for 2000 milliseconds  PORTB=0x03; // State 2 – call write function for output  PORTD=0XC0; // State 2  Delay\_ms(2000); // Delay for 2000 milliseconds  }  } |

1. **Simulate the code using PICSimLab and paste 2 screenshots showing states**

**(5p each)**

**First Screenshot**

|  |
| --- |
|  |

**Second Screenshot**

|  |
| --- |
|  |

1. **C code and simulation for Digital Output (25 Points)**
2. **Analyze the following C Language program**

**Explain each code line using comments (17 p)**

|  |
| --- |
| int mask(int character)  {  switch (character) // decide upon the following condition based on the character  {  case 0 : return 0x01; // return hex value of 1  case 1 : return 0x08; // return hex value of 8  case 2 : return 0x06; // return the hex value of 6  case 3 : return 0x30; // return the hex value of 3  }  }  void main()  { int a, x; // declare two variables ‘a’ and ‘x’  // set port A as the output and toggle it to ON  TRISA = 0x00; PORTA=0x00;  // set port D as the output and toggle it to OFF  TRISD = 0x00; PORTD=0x00;  while(1)  {  PORTA=0x20;// set port A to 32 bits  for(x=0;x<=3;x++)  { PORTD=mask(x); // set port D to a masked value of the loop counter x  Delay\_ms(2000); // delay for 2000 milliseconds  }  PORTA=0x10;// set port A to 16 bits  for(x=0;x<=3;x++)  { PORTD=mask(x); // set port D to a masked value of the loop counter x  Delay\_ms(2000);// delay 2000 milliseconds  }  }  } |
|  |

1. **Simulate the code using *PICSimLab* and paste 8 screenshots (Also draw accepted) (8p/Each 1)**

|  |
| --- |
| **First part**      **Second part** |

1. **C code and simulation for Digital Input (25 Points)**

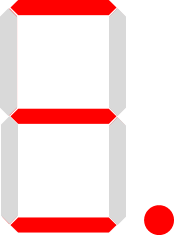
Write code for PIC 16F877A *using* C language to light PORTD bits D2, D3 and D4 if button PORTB bit4 is pressed.

When button released, all LEDs should turn off (15 p)

**Each code line should be explained using comments**  (10 p)

|  |
| --- |
| void main(){  TRISB.F3 = 1; // set 4th bit of PORTB as input  TRISD.F1 = 0; // set 2nd bit of PORTD as output    TRISD.F2 = 0; // set 3rd bit of PORTD as output  TRISD.F3 = 0; // set 4th bit of PORTD as output  // turn the LEDs OFF  PORTD.F1 = 0;  PORTD.F2 = 0;  PORTD.F3 = 0;  // in a while loop light all the LEDs  while(1) // endless while loop  {  if(PORTB.F3 == 0) // if the switch is pressed  {  Delay\_ms(100);// delay for 100 milliseconds – switch debouncing  if(PORTB.F3 == 0) // if the switch is still pressed  {  // turn the LEDS ON  PORTD.F1 = 1;  PORTD.F2 = 1;  PORTD.F3 = 1;  Delay\_ms(2000);// delay for 2 seconds    // turn the LEDs OFF  PORTD.F1 = 0;  PORTD.F2 = 0;  PORTD.F3 = 0;  }  }  }  } |
|  |

1. **C Code: Digital Input and Output (25 Points)**

Write code for PIC 16F877A using C language to light figure  on all 7 segments if button PORTB bit3 is pressed.

When button released, all 7 segments should turn off (15 p)

**Each code line should be explained using comments**  (10 p)

|  |
| --- |
| void main(){  TRISB.F2 = 1; // set 4th bit of PORTB as input    TRISD = 0x00;// set PORTD as the output  // turn the whole segment OFF  PORTD = 0x00;  // in a while loop light all the LEDs  while(1) // endless while loop  {  if(PORTB.F2 == 0) // if the switch is pressed  {  Delay\_ms(100);// delay for 100 milliseconds – switch debouncing  if(PORTB.F2 == 0) // if the switch is still pressed  {  // turn the segment ON  PORTD = 0X7F;// 0x7F is the hex value of decimal 8  Delay\_ms(1000);// delay for 2 seconds  PORTD = 0x00; // turn the segment off  }  }  }  } |

Bonus Question: (5 points)

**Analyze the following C Programming Language instructions and answer the questions:**

|  |
| --- |
| void main()  {  TRISB = 0x00;  PORTB=0x00;  while(1){  PORTB=0x03;//state 1  Delay\_ms(1500);  PORTB=0x05;//state 2  Delay\_ms(500);  }  } |

1. **What is the duration of state 1 in seconds? 1.5 seconds**
2. **What is the duration of state 2 in seconds? 0.5 seconds**