PROGRAM	: BACHELOR ENGINEERING TECHNOLOGY *ELECTRICAL ENGINEERING*
SUBJECT	: **DIGITAL TECHNOLOGY A2**
CODE	: **DIGELA2**
DATE	: MID YEAR SUPPLEMENTARY EXAMINATION JULY 2018
DURATION	: 08:00 – 11:00
WEIGHT	: 40:60
TOTAL MARKS	: 100

UNIVERSITY OF JOHANNESBURG

EXAMINER	: MR. D.R. VAN NIEKERK	720011220
MODERATOR	: JANE-ANNE BUISSON-STREET	
NUMBER OF PAGES	: 3 PAGES AND 1 ANNEXURE	

INSTRUCTIONS

1. 100 MARKS = 100%. TOTAL MARKS AVAILABLE = 100
2. ATTEMPT ALL QUESTIONS.
3. ALL DIAGRAMS AND SKETCHES MUST BE DRAWN NEATLY AND IN PROPORTION.
4. ALL DIAGRAMS AND SKETCHES MUST BE LABELLED CLEARLY.
5. ALL WORK DONE IN PENCIL, EXCEPT DIAGRAMS AND SKETCHES, WILL BE CONSIDERED AS ROUGH WORK AND WILL NOT BE MARKED.
6. MARKS WILL BE DEDUCTED FOR WORK THAT IS POORLY PRESENTED.
7. QUESTIONS MAY BE ANSWERED IN ANY ORDER, BUT ALL PARTS OF A QUESTION, MUST BE KEPT TOGETHER.
8. ONLY ONE POCKET CALCULATOR PER CANDIDATE MAY BE USED.

QUESTION 1

1.1 Which functions allow digital pins to be used as inputs or outputs and describe its associated internal pull-ups?

(5)

1.2 Which Arduino Uno pins provide external interrupts and what are the possible configurable interrupt trigger states.

(5)

1.3 Explain the problem with the following C statement, "if (x = 10)" and give a solution to this problem.

(4)

1.4 Describe one of the most common uses for a bitwise AND "&", OR "|" and XOR "^" operators.

(6)

[20]

QUESTION 2

2.1 Explain how the internal 20 KΩ pull-up resistor on the Atmega port pins, can be enabled and what will it do to an input port pin?

(5)

2.2 Generate a table showing the bit size and numeric value range of both a signed and unsigned "char", "int" and "long" data type.

(6)

2.3 Describe and explain the operation of the Arduino "shiftIn()" function?

(4)

2.4 Sketch the Arduino connected to a 74HC165, 8-bit parallel-in/serial-out shift register device. Show the clock pin (CP) connected to pin 10, the parallel load pin (PL) connected to pin 11 and the Data-out (Q7) connected to pin 12 on the Arduino. What are the values of the pull-up and down resistors and why are they required?

(8)

[23]

QUESTION 3

3.1 Write an Arduino C program that serially shifts in 8-bits of data from a 74HC165 device using the "shiftIn()" function. The clock (CP) is connected to pin 10, the Load (PL) is connected to pin 11 and the Data-Out (Q7) is connected to pin 12. Initially 8-bit data must be read continually every half second and stored into a 255 data array.

(10)

3.2 Write an Arduino C program to serial print at 9600 baud rate, a tab spaced text heading of: "CHAR DEC HEX BIN". Then print a tab spaced line with each number format for ASCII code 48 ('0') to 57 ('9') under the relevant heading on each separated line. Once the table has been printed, code execution must stop.

(12)

[22]

QUESTION 4

4.1 Without component values, sketch a high power Darlington transistor H-Bridge motor control circuit and explain how the Arduino digital pins can control the motor speed in reverse and forward directions.

(9)

4.2 Without calculating component values, sketch a H11AA1M device zero-cross detection circuit with an optically-isolated resistive triac MOC3022 device switch, to control a 230Vac, 60W lamp load. A 10k potentiometer connected to pin A0 must be used to set the triac firing delay time. The zero-crossing pulse must be detected on pin D2 and pin D5 must be used to turn the triac on.

(10)

[19]

QUESTION 5

5.1 Write an Arduino C program to detect a zero-crossing pulse on digital interrupt pin 2. After an adjustable 200us to 9407us delay that must be set by a 10k potentiometer connected to pin A0, fire an optically-isolated resistive triac switch connected to pin 5, to switch power to an AC lamp load.

(7)

5.2 What is the Arduino "servo.write()" library function used for and describing how it effects a standard and continuous rotation servo.

(9)

[16]

TOTAL [100]

Arduino Functions:

pinMode(*pin*, *mode*)
digitalWrite(*pin*, *value*)
digitalRead(pin)
analogReference(type)
analogRead(pin)
analogWrite(*pin*, *value*)
tone(pin, frequency)
tone(pin, frequency, duration)
noTone(pin)
shiftOut(dataPin, clockPin, bitOrder, value)
shiftIn(dataPin, clockPin, bitOrder)
pulseIn(pin, value)
pulseIn(pin, value, timeout)
millis()
micros()
delay(ms)
delayMicroseconds(us)
min(x, y)
max(x, y)
abs(x)
constrain(num, min, max)
map(Xin, Xmin, Xmax, Ymin, Ymax)
pow(base, exponent)
Sqrt(num)
sin(rad)
cos(rad)
tan(rad)
randomSeed(seed)
random(max)
random(min, max)
lowByte(x)
highByte(x)
bitRead(x, n)
bitWrite(x, n, b)
bitSet(x, n)
bitClear(x, n)
bit(n)
attachInterrupt(interrupt, ISR, mode)
Serial.begin(speed)
Serial.begin(speed, config)
Serial.end()
Serial.available()
Serial.flush()
Serial.print(val)
Serial.print(val, format)
Serial.println(val)
Serial.println(val, format)
Serial.read()
Serial.peek()
Serial.write(val)
Serial.write(str)
Serial.write(buf, len)
Serial.parseInt()
Serial.parseInt(char skipChar)
Serial.parseFloat()
Serial.readBytes(buffer, length)
Serial.readBytesUntil(terminator, buffer, length)
Serial.readString()
Serial.readStringUntil(terminator)
Serial.find(target)
Serial.findUntil(target, terminator)
Serial.setTimeout(time)
serialEvent()
LiquidCrystal lcd(rs, en, d4, d5, d6, d7)
LiquidCrystal lcd(rs, rw, en, d0, d1, d2, d3, d4, d5, d6, d7)
lcd.begin(cols, rows)
lcd.clear()
lcd.home()
lcd.setCursor(col, row)
lcd.write(byte)
lcd.print(data)
lcd.print(data, base)
lcd.cursor()
lcd.noCursor()
lcd.blink()
lcd.noBlink()
lcd.display()
lcd.noDisplay()
lcd.scrollDisplayLeft()
lcd.scrollDisplayRight()
lcd.leftToRight()
lcd.rightToLeft()
lcd.autoscroll()
lcd.noAutoscroll()
lcd.createChar(num, data)
EEPROM.read(address)
EEPROM.write(address, value)
EEPROM.update(address, value)
EEPROM.put(address, data)
EEPROM.get(address, data)
EEPROM[address]
servoM1.attach(pin)
servoM1.attach(pin, min, max)
servo.write(angle)
servo.writeMicroseconds(us)
servo.read()
servo.attached()
servo.detach()
NumKeys.begin(makeKeymap(keys));
NumKeys.setDebounceTime(time);
NumKeys.setHoldTime(time);
char ckey = NumKeys.getKey()
byte State = NumKeys.getState()
boolean State = NumKeys.keyStateChanged()
char ckey = NumKeys.waitForKey()
NumKeys.addEventListener(keypadEvent)