



UNIVERSITY COLLEGE TATI (UCTATI)

FINAL EXAMINATION QUESTION BOOKLET

COURSE CODE	: BMT 4033
COURSE TITLE	: EMBEDDED SYSTEM DESIGN
SEMESTER/SESSION	: 1-2023/2024
DURATION	: 3 HOURS

Instructions:

1. This booklet contains **4** questions. Answer **all questions**.
2. All theory and calculation answers should be written in the answer booklet.
3. Write legibly and draw sketches wherever required.
4. If in doubt, raise your hands and ask the invigilator.

DO NOT OPEN THIS BOOKLET UNTIL YOU ARE TOLD TO DO SO

THIS BOOKLET CONTAINS 10 PRINTED PAGES INCLUDING COVER PAGE

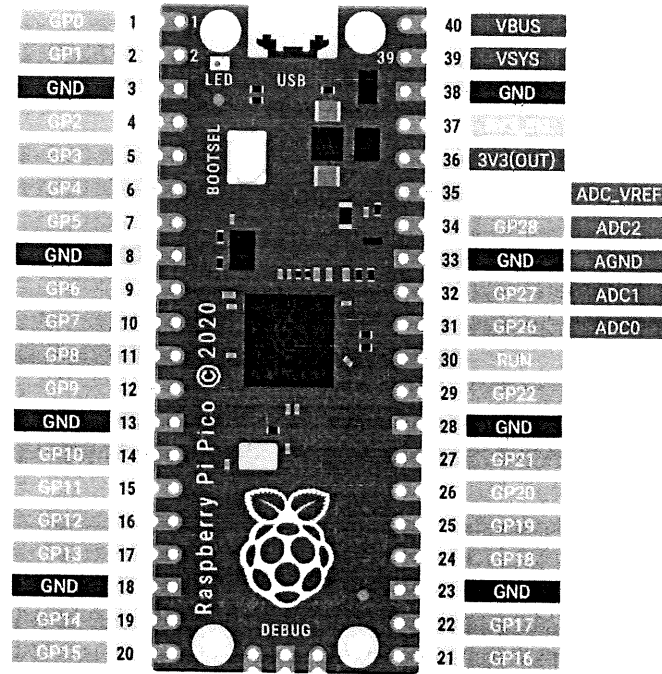


Figure 2: Pi Pico pin out.



UNIVERSITY COLLEGE TATI (UCTATI)

FINAL EXAMINATION QUESTION BOOKLET

COURSE CODE	: BMT 4033
COURSE TITLE	: EMBEDDED SYSTEM DESIGN
SEMESTER/SESSION	: 1-2023/2024
DURATION	: 3 HOURS

Instructions:

1. This booklet contains **4** questions. Answer **all questions**.
2. All theory and calculation answers should be written in the answer booklet.
3. Write legibly and draw sketches wherever required.
4. If in doubt, raise your hands and ask the invigilator.

DO NOT OPEN THIS BOOKLET UNTIL YOU ARE TOLD TO DO SO

THIS BOOKLET CONTAINS 10 PRINTED PAGES INCLUDING COVER PAGE

QUESTION 1

- a) **Describe** the difference between python language with c language in developing embedded applications by giving **five (5)** comparisons. (5 marks)
- b) **Figure out** a flow chart based on the program below.

```
from machine import ADC,Pin
from utime import sleep

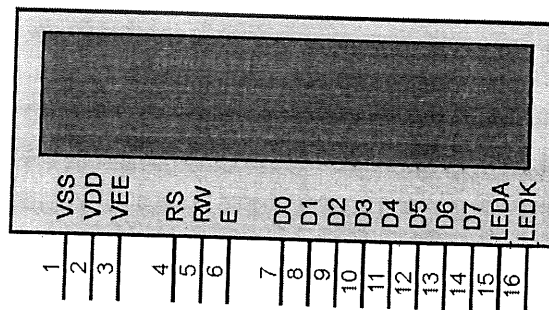
pot = ADC(Pin(27))      # create ADC object on ADC pin
led1 = Pin(3, Pin.OUT)  #create led output at pin 3
led2 = Pin(12, Pin.OUT) #create led output at pin 12
while 1:
    value=pot.read_u16() # read adc put in value
    print("raw value:",value)
    volt=value/65535*3.300;
    print("voltage",volt)
    if volt<1:
        led1.value(0) # led is off
        led2.value(0)
    elif volt<2:
        led1.value(1) # led is off
        led2.value(0)
    else:
        led1.value(1) # led is on
        led2.value(1)
    sleep(0.5)
```

(15 marks)

QUESTION 2

- a) Figure 1 shows the pin out of the 16x2 LCD. **Describe** the function of VSS, VDD, VEE, RS, RW, E and data bus (D0-D7) at LCD.

(7 marks)

**Figure 1: LCD pin out**

- b) The Raspberry Pi Pico has 3 analog input pins, which convert an analog input (between 0-3.3V by default) into a digital value with 16 bits of resolution. Create an application which **flashing LED change** based on the **setting of the potentiometer** which minimum time is 0.5 sec and maximum time is 2.5 sec. Use a bank of 4 LEDs and potentiometer reading as ADC input. Connect each LEDs to an output of the Raspberry Pi Pico. Make your code to make all the LEDs flash together.

- i. **Produce** schematic diagram for LED connection. Refer to figure 2 for the raspberry pi Pico pin out.

(4 marks)

- ii. **Identify** a program for this application.

(6 marks)

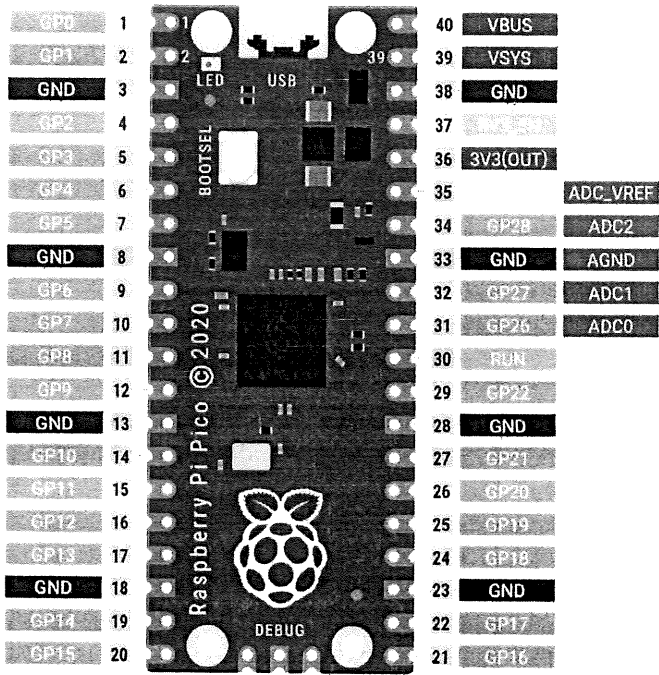


Figure 2: Pi Pico pin out.

QUESTION 3

- a) Draw the internal construction of the 4x4 matrix keypad. (5 marks)
- b) Create an application that will give output at LCD 16x2 using **4-bit mode** connection based on the keypad input value as follows:
- When the numbering value is pressed, the value is displayed on the first line at the center of the LCD.
 - When * is pressed, it will clear the LCD screen.
 - When either A, B, C, D or # is pressed, it will display a different message at the LCD screen at the second line of LCD.
Example: if A is pressed, it will display "welcome".
if B is pressed, it will display "thank you".
- i. Referring to the LCD and keypad in figure 3, **produce** the wiring diagram between these devices and the Pi Pico microcontroller. For Pi Pico pin out, refer to figure 2 in question 2. (10 marks)
- ii. **Identify** a program for this application. Refer attachment section for the LCD and keypad basic program as reference. (15 marks)

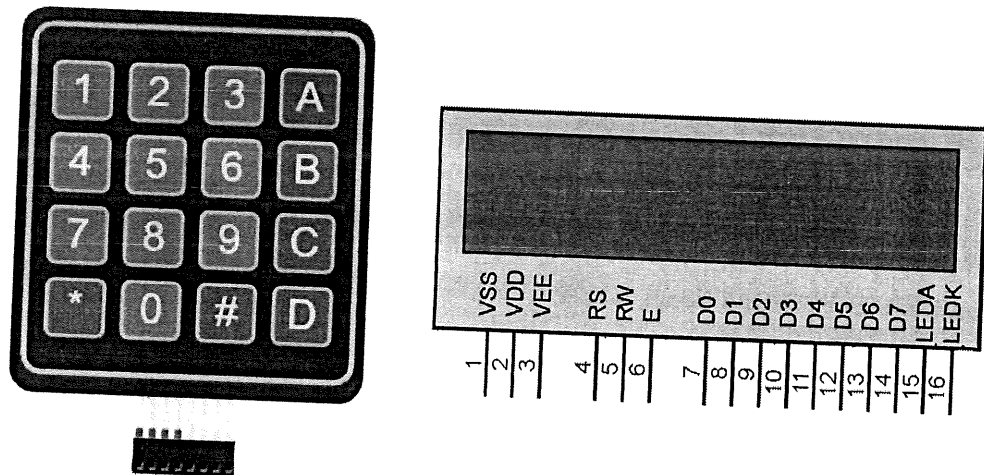


Figure 3: keypad and LCD

QUESTION 4

- a) **State** the function of a 7-segment display unit, then **identify types** of 7 segment display unit and **describe the difference** of these display types. (5 marks)
- b) Using four (4) 7-segment displays in TM1637 module, create an application which reads data from SR04 ultrasonic distance sensor in centimeter, then display it on 7 segments. Only integer value is displayed.
- i) **Produce** the schematic diagram for the connection between Pi Pico, SR04 ultrasonic distance sensor and TM1637 module. Refer figure 4 for Tm1637 module, figure 5 for SR04 ultrasonic distance sensor. (15 marks)
- ii) **Identify** a program for this application. (13 marks)

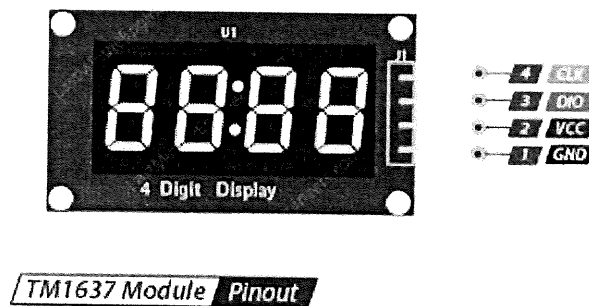


Figure 4: TM1637 module

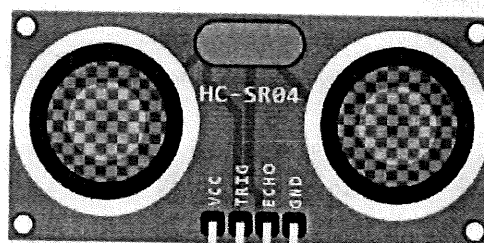


Figure 5: SR04 ultrasonic distance sensor

-----End of question-----

Attachment:

LCD.py 4-bit mode basic program routine example

```

from machine import Pin
from utime import sleep

rs=Pin(2,Pin.OUT)
e=Pin(3,Pin.OUT)
data=[Pin(4,Pin.OUT),Pin(5,Pin.OUT),Pin(6,Pin.OUT),Pin(7,Pin.OUT)]
      #   db4           db5           db6           db7
def write(nilai): #function to write value to lcd
    data[0].value((nilai & 0x10)>>4)      #send high nibble
    data[1].value((nilai & 0x20)>>5)
    data[2].value((nilai & 0x40)>>6)
    data[3].value((nilai & 0x80)>>7)
    e.value(0)          #enable low
    sleep(0.01)
    e.value(1)          #enable high
    sleep(0.01)

    data[0].value((nilai & 0x01)>>0)      #send low nibble
    data[1].value((nilai & 0x02)>>1)
    data[2].value((nilai & 0x04)>>2)
    data[3].value((nilai & 0x08)>>3)
    e.value(0)          #enable low
    sleep(0.01)
    e.value(1)          #enable high
    sleep(0.01)

def cmd_write(nilai): #function to write command
    rs.value(0)        #set rs 0
    write(nilai)
    sleep(0.01)

def data_write(nilai): #function to write data
    rs.value(1)        #set rs 1
    write(nilai)

def lcd_start():      #function to start lcd
    cmd_write(0x02)
    sleep(0.2)
    cmd_write(0x28)
    sleep(0.05)
    cmd_write(0x0c)
    cmd_write(0x01)
    cmd_write(0x80)

```

```

def print_lcd(string): #print at lcd
    for i in string:
        data_write(ord(i)) # pecah word ke satu2 character

def zfl(s, width):
    # Pads the provided string with leading 0's to suit the specified 'chrs'
    length
    # Force # characters, fill with leading 0's
    return '{:0>{w}}'.format(s, w=width)

```

Code (Hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Decrement cursor (shift cursor to left)
6	Increment cursor (shift cursor to right)
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning to 1st line
C0	Force cursor to beginning to 2nd line
38	2 lines and 5x7 matrix

LCD command code

Keypad.py basic program routine example

```
from machine import Pin
from utime import sleep

#           row 1           row 2           row 3           row 4
key_row=[Pin(26, Pin.OUT),Pin(22, Pin.OUT),Pin(21, Pin.OUT),Pin(20,
Pin.OUT)]

#           col 1           col 2
key_col=[Pin(19, Pin.IN, Pin.PULL_UP),Pin(18, Pin.IN, Pin.PULL_UP),
#           col 3           col 4
Pin(17, Pin.IN, Pin.PULL_UP),Pin(16, Pin.IN, Pin.PULL_UP)]

key_label=['7','8','9','/',
           '4','5','6','x',
           '1','2','3','- ',
           '0','0','=','+']

def getkey(): # function to read keypad then return value pressed
    for i in range(4): # set all row to 1
        key_row[i].value(1)
    for i in range(4):
        key_row[i].value(0) # set row to 0
        for j in range(4):
            if(key_col[j].value()==0): #read
                sleep(0.1) # delay
                if(key_col[j].value()==0):#read again
                    while(key_col[j].value()==0): #wait key depressed
                        pass # pass while function
                    return key_label[j+i*4] #return key pressed
        key_row[i].value(1) # set row to 1
    return 0 # return no key pressed
```

Tm1637 example program

```
import tm1637
from machine import Pin
from time import sleep
tm = tm1637.TM1637(clk=Pin(0), dio=Pin(1))
sw1 = Pin(9, Pin.IN, Pin.PULL_UP) # enable internal pull-up resistor
number=0
while 1:
    sw1_value=sw1.value()
    if sw1_value==0:
        number=number+1
        if number>9999:
            number=9999
        print(number)
        sleep(0.4)
    tm.number(number)
```

ADC program routine example

```
from machine import ADC, Pin
adc = ADC(Pin(26))      # create ADC object on ADC pin
adc.read_u16()          # read value, 0-65535 across voltage range 0.0v -
3.3v
```

SR04 ultrasonic sensor program example

```
from machine import Pin,time_pulse_us
from time import sleep,sleep_us
echo = Pin(19, Pin.IN)
trigger = Pin(20, Pin.OUT)
while 1:
    trigger.value(1)
    sleep_us(10)
    trigger.value(0)
    timing=time_pulse_us(echo,1,1000000)
    if timing<0:
        print("error measurement")
    else:
        distance=timing/58
        print("distance",distance,"cm")
    sleep(0.5)
```