

**UNIVERSITY COLLEGE TATI (UC TATI)****FINAL EXAMINATION QUESTION BOOKLET**

COURSE CODE	: DMT 3073 / DEI 3073
COURSE	: APPLICATION OF MICROCONTROLLER
SEMESTER/SESSION	: 1 - 2023/2024
DURATION	: 3 HOURS

Instructions:

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

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

THIS BOOKLET CONTAINS 9 PRINTED PAGES INCLUDING COVER PAGE

QUESTION 1

- a) Name the microcontroller chip used in Arduino Uno. (1 mark)
- b) State the total number of digital pins and analog pins in Arduino Uno. (2 marks)
- c) State the crystal oscillator frequency used at Arduino Uno board. (1 mark)
- d) Label the following component on Arduino Uno board shown on Figure 1: (5 marks)

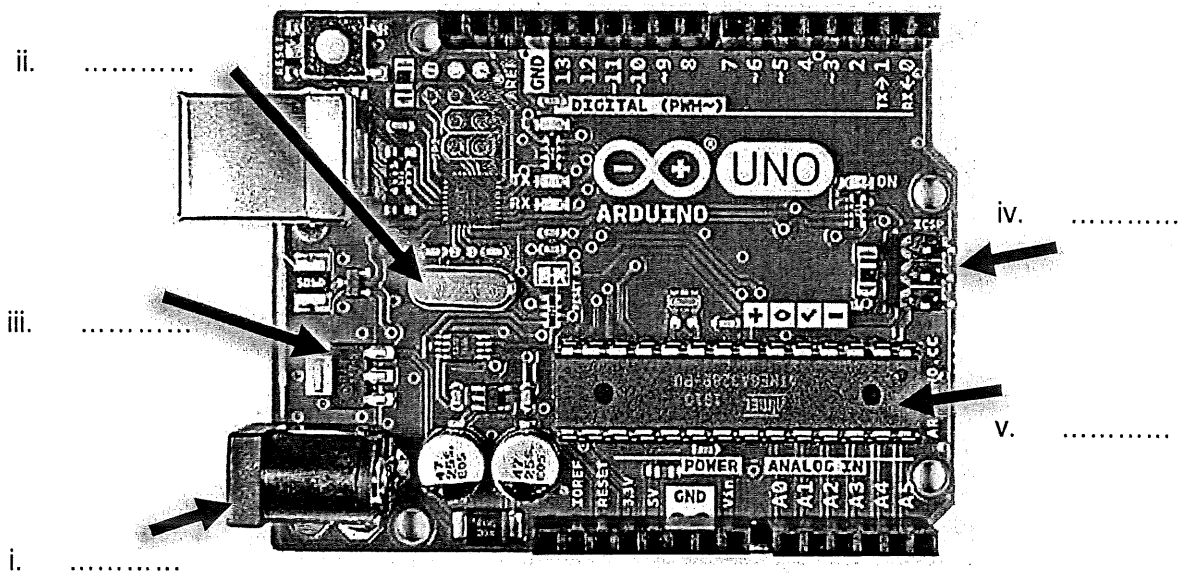


Figure 1

- e) Name all 'TIMER' register in Atmel ATMEGA328 (3 marks)
- f) Compare the features of all 'TIMER' register in Atmel ATMEGA328 in term of size (bits) and interrupt. (6 marks)

QUESTION 2

- a) Produce a schematic diagram consist of 4 LED at pin 10,11,12 and 13 of Arduino and bluetooth module (HC-06) at pin 2 (Rx) and pin 3 (Tx). **Use Appendix A to answer this question and attach it with answer booklet. (At the end of question paper).**
(5 marks)
- b) Produce a program to turn ON/OFF LED according to the schematic drawn in question 2 (a) and smartphone application in Figure 2 when:
- Button icon 'A' is pressed (touch), the LEDs ON simultaneously.
 - When button 'B' is pressed (touch), all LED is OFF.

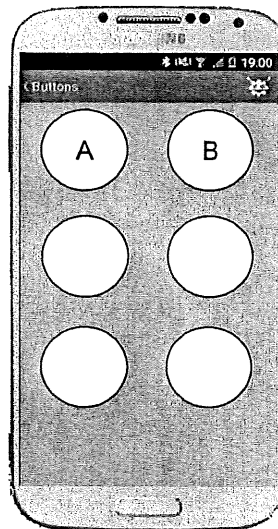


Figure 2

(13 marks)

- c) Produce a schematic diagram by using one 7-segment and 7-segment decoder (7447) control by Arduino microcontroller. It also has one (1) pushbutton to trigger the 7-segment output. Assign any pin as input/output. **Use Appendix B to answer this question and attach it with answer booklet. (At the end of question paper).**
(4 marks)
- d) Based on schematic drawn at question 2 c), when the pushbutton pressed once, produce a program to:
- Count-up the 7-segment starting from zero '0' to nine '9' then,
 - Count-down from nine '9' to zero '0'.

The process can be start again after finish the previous cycle.

(19 marks)

APPLICATION OF MICROCONTROLLER (DMT 3073 / DEI 3073)

QUESTION 3

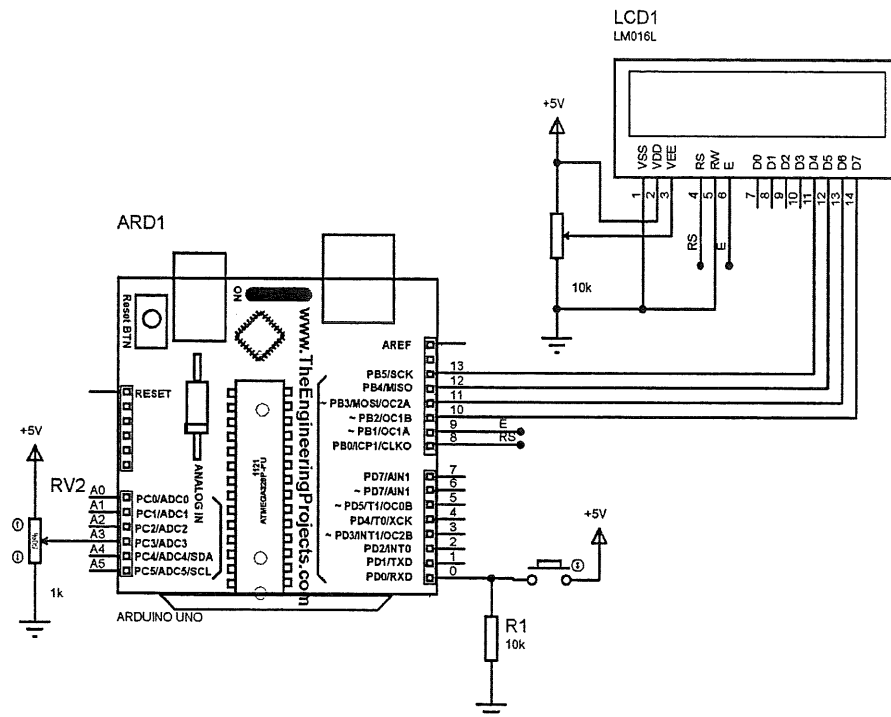


Figure 3

- a) Based on schematic given in Figure 3, produce a program to display:
- MICROCONTROLLER at upper row. (5 marks)
 - Examination at lower row after pressing the pushbutton once. (7 marks)
- b) Based on schematic given in Figure 3, produce a program to:
- Display the "Analog read =" at upper row. (5 marks)
 - Read and display the potentiometer value at lower row. (7 marks)

QUESTION 4

- a) Produce a schematic diagram which control a 12V DC motor using 5V relay and the relay coil is controlled by ULN 2803 IC through Arduino microcontroller. The direction (forward / reverse) of the motor is controlled by using H-Bridge connection and triggered by using two pushbuttons for forward and reverse direction. **Use Appendix C to answer this question and attach it with answer booklet.** *(At the end of question paper).*

(6 marks)

- b) Based on schematic drawn at question 4 a), produce a program to:
- Turn ON DC motor (forward) for 2 second when pushbutton 1 pressed.
 - Turn ON DC motor(reverse) for 2 second when pushbutton 2 pressed.

It must be no conflict between forward and reverse direction.

(11 marks)

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

ATTACHMENT 1: ARDUINO CHEAT SHEET

ARDUINO CHEAT SHEET V.02B

Mostly taken from the extended reference:
<http://arduino.cc/en/Reference/Extended>
 Gavin Smith – Robots and Dinosaurs, The Sydney Hackspace

Flash (2K for bootloaders)		EEPROM	
ATmega168	16KB	ATmega328	1KB
ATmega328P	32KB	ATmega328P	4KB
ATmega328P	32KB	ATmega328P	4KB

Pin	Function	Pin	Function
0	GND	13	GND
1	5V	14	5V
2	Digital In/Out	15	Digital In/Out
3	Digital In/Out	16	Digital In/Out
4	Digital In/Out	17	Digital In/Out
5	Digital In/Out	18	Digital In/Out
6	Digital In/Out	19	Digital In/Out
7	Digital In/Out	20	Digital In/Out
8	Digital In/Out	21	Digital In/Out
9	Digital In/Out	22	Digital In/Out
10	Digital In/Out	23	Digital In/Out
11	Digital In/Out	24	Digital In/Out
12	Digital In/Out	25	Digital In/Out

Pin	Function	Pin	Function
0	GND	13	GND
1	5V	14	5V
2	Digital In/Out	15	Digital In/Out
3	Digital In/Out	16	Digital In/Out
4	Digital In/Out	17	Digital In/Out
5	Digital In/Out	18	Digital In/Out
6	Digital In/Out	19	Digital In/Out
7	Digital In/Out	20	Digital In/Out
8	Digital In/Out	21	Digital In/Out
9	Digital In/Out	22	Digital In/Out
10	Digital In/Out	23	Digital In/Out
11	Digital In/Out	24	Digital In/Out
12	Digital In/Out	25	Digital In/Out

Pin	Function	Pin	Function
0	GND	13	GND
1	5V	14	5V
2	Digital In/Out	15	Digital In/Out
3	Digital In/Out	16	Digital In/Out
4	Digital In/Out	17	Digital In/Out
5	Digital In/Out	18	Digital In/Out
6	Digital In/Out	19	Digital In/Out
7	Digital In/Out	20	Digital In/Out
8	Digital In/Out	21	Digital In/Out
9	Digital In/Out	22	Digital In/Out
10	Digital In/Out	23	Digital In/Out
11	Digital In/Out	24	Digital In/Out
12	Digital In/Out	25	Digital In/Out

Structure

```
void setup() {
  // initialization
}

void loop() {
  // main loop
}
```

Control Structures

```
if (x < y) { } else { }
switch (myvar) {
  case 1:
    break;
  case 2:
    break;
  default:
    break;
}
for (int i=0; i<=255; i++) { }
while (x < y) { }
do { } while (x < y);
continue; // Go to next in do/while loop
return x; // Or 'return;' for void.
goto // considered harmful :-)
```

Further Syntax

```
// (single line comment)
/* (multi-line comment) */
#define DOZEN 12 // Not balck 's!
#include <avr/pgmspace.h>
```

General Operators

```
= (assignment operator)
+ (addition)
- (subtraction)
* (multiplication)
/ (division)
% (modulo)
++ (equal to)
-- (less than)
> (greater than)
< (less than or equal to)
>= (greater than or equal to)
<= (less than or equal to)
&& (and)
|| (or)
! (not)
```

Pointer Access

```
& reference operator
* dereference operator
```

Bitwise Operators

```
& (bitwise and)
| (bitwise or)
^ (bitwise xor)
~ (bitwise not)
<< (bitshift left)
>> (bitshift right)
```

Compound Operators

```
++ (increment)
-- (decrement)
+= (compound addition)
-= (compound subtraction)
*= (compound multiplication)
/= (compound division)
&= (compound bitwise and)
|= (compound bitwise or)
```

Qualifiers

```
static // persists between calls
volatile // use RAM (nice for ISR)
const // make read-only
PROGRAM // use flash
```

Digital I/O

```
pinMode(pin, INPUT/OUTPUT)
digitalWrite(pin, value)
int digitalWrite(pin)
// Write High to inputs to use pull-up res
```

Analog I/O

```
analogReference(DEFAULT/INTERNAL/EXTERNAL)
int analogRead(pin) // Call twice if switching pins from high Z source.
analogWrite(pin, value) // PWM
```

Advanced I/O

```
tone(pin, freqHz)
tone(pin, freqHz, duration_ms)
noTone(pin)
shiftOut(dataPin, clockPin, MSBFIRST/LSBFIRST, value)
unsigned long pulseIn(pin, [HIGH/LOW])
```

Time

```
unsigned long millis() // 50 days overflow
unsigned long micros() // 70 min overflow
delay(ms)
delayMicroseconds(us)
```

Math

```
min(x, y)
max(x, y)
abs(x)
constrain(x, minval, maxval)
map(val, fromL, fromH, toL, toH)
pow(base, exponent)
sqrt(x)
sin(rad)
cos(rad)
tan(rad)
```

Random Numbers

```
randomSeedSeed() // Long or int
long random(max)
long random(min, max)
```

Bits and Bytes

```
lowByte()
highByte()
bitRead(x, bit)
bitWrite(x, bit, val)
bitSet(x, bit)
bitClear(x, bit)
bit(bint) // bint: 0-LSB 7-MSB
```

Strings

```
char S[11];
char S2[8] = "a,b,c,d,e,f,g,h,i,j,k";
char S3[8] = "a,b,c,d,e,f,g,h,i,j,k";
// Included \0 null termination
char S4[] = "arduino";
char S5[8] = "arduino";
char S6[15] = "arduino";
```

Arrays

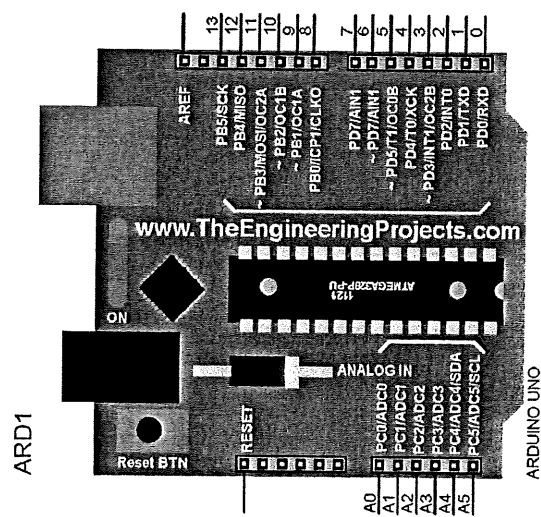
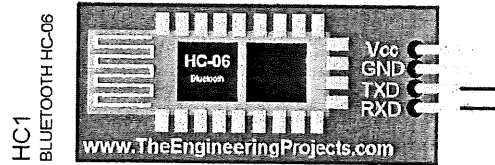
```
int myInts[6];
int myPins[12] = {2, 4, 8, 3, 6};
int mySensorVals[6] = {2, 4, 8, 3, 2};
```

Conversion

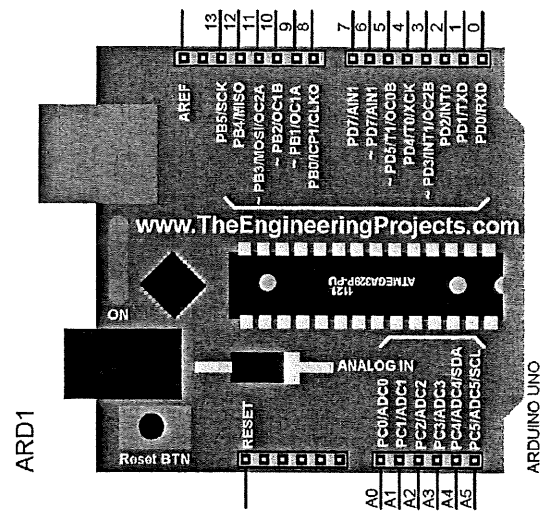
```
char[]
byte[]
word[]
float[]
int[]
long[]
```

APPLICATION OF MICROCONTROLLER (DMT 3073 / DEI 3073)

APPENDIX A: Answer for question 2 a)



APPENDIX B: Answer for question 2 c)



APPLICATION OF MICROCONTROLLER (DMT 3073 / DEI 3073)

APPENDIX C: Answer for question 4 a)

