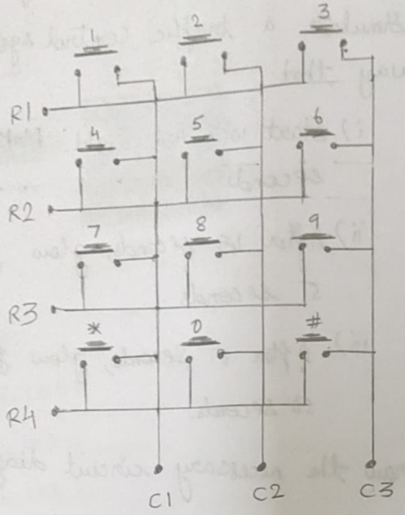
1) 4 × 3 KEYPAD CIRCUIT ARRANGEMENT



EX. NO: 03
DATE: 04-02-2024

KEYPAD INTERFACING

AIM:

To interface 4x3 Keypad with Arduino and to access it without and with using Keypad.h library.

COMPONENTS REQUIRED:

Arduino UNO, 4x3 keypad module, jumper, Arduino IDE.

PRELAB QUESTIONS:

1) Draw the electronic circuit implemented inside 4x3 keypad.

2) How many pins does 4x3 Keypad have? Explain them.

A 4x3 keypad has 7 pins. They are arranged in a matrix form, with 4 rows and 3 columns. Each row pin is connected to a set of keys in the same row and each column pin is connected to a set of ~~rows~~ Keys in the same column.

4) Step 1: Set C1 to low

```
0 1 1 → HIGH
0 [1] 1 → LOW
0 1 1 → HIGH
0 1 1 → HIGH
```

Set^ C1 to HIGH

```
1 1 1
1 [1] 1
1 1 1
1 1 1
```

Step 2: Step C2 to low

```
1 0 1 → HIGH
1 [0] 1 → LOW
1 0 1 → HIGH
1 0 1 → HIGH
```

* 2nd row is LOW

* R2 and C2 Keys

* The key is 5.

3) Explain the mapping of value with row and column for all 12 keys.

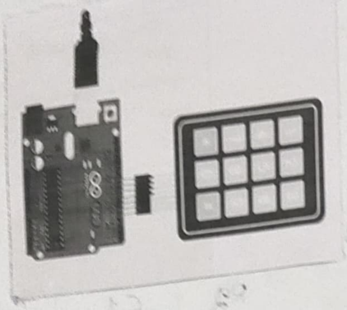
KEY	ROW #	COL #
1	R1	C1
2	R1	C2
3	R1	C3
4	R2	C1
5	R2	C2
6	R2	C3
7	R3	C1
8	R3	C2
9	R3	C3
*	R4	C1
0	R4	C2
#	R4	C3

4) When a key is pressed in the keypad what happens to the signal level of various rows and col. from pin 1 to 7. Illustrate this scenario for the Key press 5.

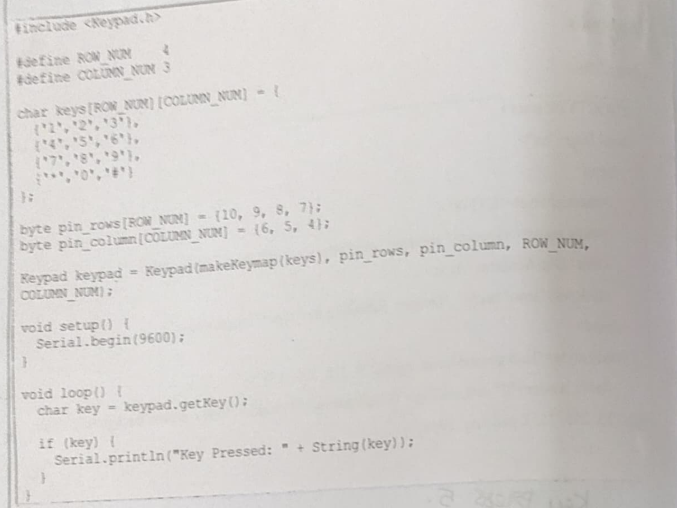
PROCESS:

```
for each column {
    set column pin low;
    for each row {
        if row pin is low {
            Key is pressed;
            Determine which key;
        }
    }
    set column pin high;
}
```

1) CIRCUIT DIAGRAM:



PROGRAM:



```
#include <Keypad.h>

#define ROW_NUM     4
#define COLUMN_NUM 3

char keys[ROW_NUM][COLUMN_NUM] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};

byte pin_rows[ROW_NUM] = {10, 9, 8, 7};
byte pin_column[COLUMN_NUM] = {6, 5, 4};

Keypad keypad = Keypad(makeKeymap(keys), pin_rows, pin_column, ROW_NUM,
COLUMN_NUM);

void setup() {
  Serial.begin(9600);
}

void loop() {
  char key = keypad.getKey();

  if (key) {
    Serial.println("Key Pressed: " + String(key));
  }
}
```

5) Can the interfacing of Keypad be done without the header file <Keypad.h>? If so, how?

Yes, it is possible.

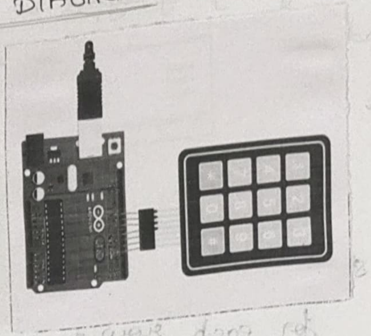
STEPS:

```
for each column {
    set column pin low ~~&~~
        for each row {
            if row pin is low {
                    Determine the key based on
the row and the column;
            }  }
    set column pin high;
}
```

EXERCISE:

1) Identify the key stroke and display it in the serial monitor using Library.

2) CIRCUIT DIAGRAM:

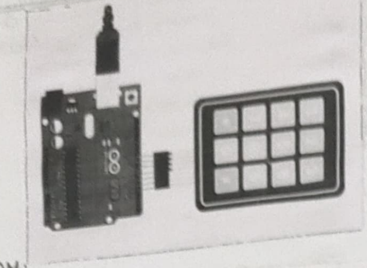


PROGRAM:

```
#define ROW_NUM     4
#define COLUMN_NUM 3

char keys[ROW_NUM][COLUMN_NUM] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};

byte pin_rows[ROW_NUM] = {10, 9, 8, 7};
byte pin_column[COLUMN_NUM] = {6, 5, 4};

void setup() {
  Serial.begin(9600);
  for (byte i = 0; i < ROW_NUM; i++) {
    pinMode(pin_rows[i], INPUT_PULLUP);
  }
  for (byte i = 0; i < COLUMN_NUM; i++) {
    pinMode(pin_column[i], OUTPUT);
    digitalWrite(pin_column[i], HIGH);
  }
}

void loop() {
  char key = getKey();
  if (key != -1) {
    Serial.println(key);
    delay(500); // Add a small delay to debounce the keypress
  }
}

char getKey() {
  for (byte i = 0; i < COLUMN_NUM; i++) {
    digitalWrite(pin_column[i], LOW);
    for (byte j = 0; j < ROW_NUM; j++) {
      if (digitalRead(pin_rows[j]) == LOW) {
        delay(50); // Add a small delay to debounce the keypress
        while (digitalRead(pin_rows[j]) == LOW); // Wait for the key to be
released
        digitalWrite(pin_column[i], HIGH);
        return keys[j][i];
      }
    }
    digitalWrite(pin_column[i], HIGH);
  }
  return -1; // No key pressed
}
```

2) Identify the keystroke and display it in the serial monitor without using library.

3) CIRCUIT DIAGRAM:



PROGRAM:

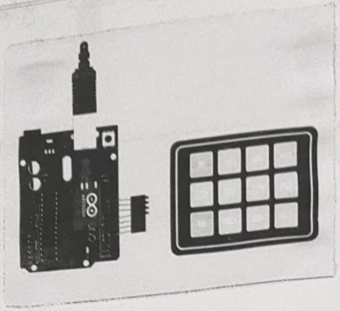
```
#define ROW_NUM     4
#define COLUMN_NUM 3

char keys[ROW_NUM][COLUMN_NUM] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};

byte pin_rows[ROW_NUM] = {10, 9, 8, 7};
byte pin_column[COLUMN_NUM] = {6, 5, 4};
byte ledPin = 13;

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);

  for (byte i = 0; i < ROW_NUM; i++) {
    pinMode(pin_rows[i], INPUT_PULLUP);
  }
  for (byte i = 0; i < COLUMN_NUM; i++) {
    pinMode(pin_column[i], OUTPUT);
    digitalWrite(pin_column[i], HIGH);
  }
}

void loop() {
  char key = getKey();
  if (key != 0) {
    if (key >= '0' && key <= '9') {
      int num = key - '0';
      blinkLED(num);
    }
    delay(500); // Add a small delay to debounce the keypress
  }
}

void blinkLED(int count) {
  for (int i = 0; i < count; i++) {
    digitalWrite(ledPin, HIGH);
    delay(500);
    digitalWrite(ledPin, LOW);
    delay(500);
  }
}

char getKey() {
  for (byte i = 0; i < COLUMN_NUM; i++) {
    digitalWrite(pin_column[i], LOW);
    for (byte j = 0; j < ROW_NUM; j++) {
      if (digitalRead(pin_rows[j]) == LOW) {
        delay(50); // Add a small delay to debounce the keypress
        while (digitalRead(pin_rows[j]) == LOW); // Wait for the key to be
released
        digitalWrite(pin_column[i], HIGH);
        return keys[j][i];
      }
    }
    digitalWrite(pin_column[i], HIGH);
  }
  return 0; // No key pressed
}
```

3) Identify the number pressed in the keypad and make the in-built LED glow for the value of the number.

i) LED initially must be in LOW

ii) If number pressed is 'x', LED must blink 'x' times then go to LOW state.

1) CIRCUIT DIAGRAM:



PROGRAM:

```
#include <Keypad.h>

#define ROW_NUM     4
#define COLUMN_NUM 3

char keys[ROW_NUM][COLUMN_NUM] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};

byte pin_rows[ROW_NUM] = {10, 9, 8, 7};
byte pin_column[COLUMN_NUM] = {6, 5, 4};

Keypad keypad = Keypad(makeKeymap(keys), pin_rows, pin_column, ROW_NUM,
COLUMN_NUM);

String inputBuffer = "";

void setup() {
  Serial.begin(9600);
}

void loop() {
  char key = keypad.getKey();

  if (key) {
    Serial.println("Key Pressed: " + String(key));

    if (key >= '0' && key <= '9') {
      inputBuffer += key;
    } else if (key == '#') {
      if (inputBuffer.length() > 0) {
        int inputValue = inputBuffer.toInt();
        if (inputValue >= 0 && inputValue <= 255) {
          Serial.println("Valid Input: " + String(inputValue));

          char asciiValue = inputValue;
          Serial.println("ASCII Value: " + String(asciiValue));
        } else {
          Serial.println("Invalid Input. Input should be between 0 and 255.");
        }
        inputBuffer = "";
      } else {
        Serial.println("Invalid Input. Input is empty.");
      }
    }
  }
}
```

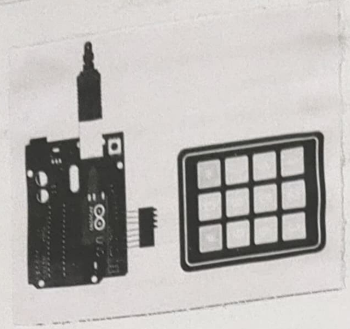
POST-LAB:

1) Write an Arduino sketch

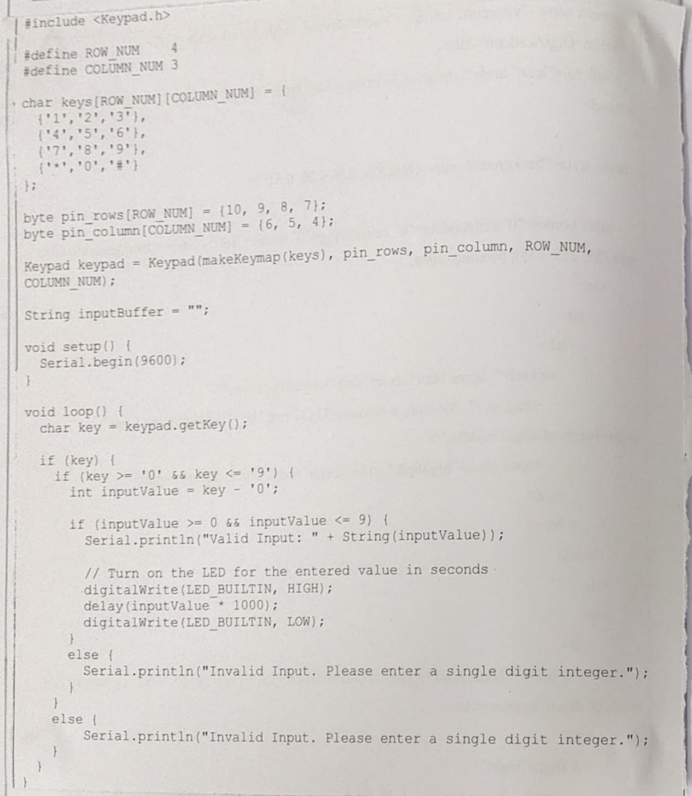
a) Scan the input in the Keypad

b) Convert the ASCII to integer and display in the serial monitor.

2) CIRCUIT DIAGRAM.



PROGRAM:



```
#include <Keypad.h>

#define ROW_NUM    4
#define COLUMN_NUM 3

char keys[ROW_NUM][COLUMN_NUM] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};

byte pin_rows[ROW_NUM] = {10, 9, 8, 7};
byte pin_column[COLUMN_NUM] = {6, 5, 4};

Keypad keypad = Keypad(makeKeymap(keys), pin_rows, pin_column, ROW_NUM,
COLUMN_NUM);

String inputBuffer = "";

void setup() {
  Serial.begin(9600);
}

void loop() {
  char key = keypad.getKey();

  if (key) {
    if (key >= '0' && key <= '9') {
      int inputValue = key - '0';

      if (inputValue >= 0 && inputValue <= 9) {
        Serial.println("Valid Input: " + String(inputValue));

        // Turn on the LED for the entered value in seconds
        digitalWrite(LED_BUILTIN, HIGH);
        delay(inputValue * 1000);
        digitalWrite(LED_BUILTIN, LOW);
      }
      else {
        Serial.println("Invalid Input. Please enter a single digit integer.");
      }
    }
    else {
        Serial.println("Invalid Input. Please enter a single digit integer.");
    }
  }
}
```

2) Write an Arduino Sketch

a) Scan the input in the keypad

b) Extract the value. The inbuilt LED must be in HIGH State for the 'x' seconds and in Low otherwise.

RESULT:

Thus, 4x3 Keypad interfacing with Arduino with and without using library is done