

ENGINE

Teaching online electronics, microcontrollers and programming in Higher Education

Output 2: Online Course for Microcontrollers: syllabus, open educational resources

Practice leaflet: Module_1-5 Keypad 4x4

Lead Partner: International Hellenic University (IHU)

Authors: Theodosios Sapounidis [IHU], Aristotelis Kazakopoulos [IHU], Aggelos Giakoumis [IHU], Sokratis Tselegkaridis [IHU]

Declaration

This report has been prepared in the context of the ENGINE project. Where other published and unpublished source materials have been used, these have been acknowledged.

Copyright

© Copyright 2021 - 2023 the [ENGINE](#) Consortium

Warsaw University of Technology (Poland)

International Hellenic University (IHU) (Greece)

European Lab for Educational Technology- EDUMOTIVA (Greece)

University of Padova (Italy)

University of Applied Sciences in Tarnow (Poland)

All rights reserved.



This document is licensed to the public under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

Funding Disclaimer

This project has been funded with support from the European Commission. This report reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Table of Contents

Executive summary	4
Chapter 1: Overview	5
Chapter 2: Activities	8
2.1 Activity 1. Basics with keypad 4x4	8
2.2 Activity 2. Keypad 4x4 and LCD 16x2	12
2.3 Activity 3. Simple calculator	16
Chapter 3: Recapitulation	22
References	23

Executive summary

In this Module we will use a Keypad 4x4.

Chapter 1: Overview

Table 1. Overview

Title / short summary	Keypad 4x4
Expected learning outcomes	<p>Students completing the course will be able to:</p> <ul style="list-style-type: none">• Recognize basic Arduino Uno functions and programming structures• Use a library to communicate with a keypad 4x4• Design and implement simple circuits with a keypad 4x4
Keywords	Keypad 4x4
Duration	<p>The duration of the module_1-5 is 3 hours</p> <ul style="list-style-type: none">• Module_1-5 slides - 30 minutes• 1st activity: Basics with keypad 4x4 - 50 minutes• 2rd activity: Keypad 4x4 and LCD 16x2 - 50 minutes• 3rd activity: Simple calculator - 50 minutes

Involved	<p>The students:</p> <ul style="list-style-type: none"> • Take part in activities • Answer questionnaires <p>The teachers:</p> <ul style="list-style-type: none"> • Show the presentation of the module • Answer questions • Point out the tips • Encourage participation and discussion
Assignment	<p>The module_1-5 includes:</p> <ul style="list-style-type: none"> • 2 Open Projects
Educational tools and equipment	<ul style="list-style-type: none"> • Material: PC • Software: browser, Tinkercad
Prerequisites / pre-existing knowledge	<ul style="list-style-type: none"> • Students should have knowledge of wiring electronic components in breadboard (link1) • Students should have basic programming knowledge in C language (link2) • Students should be familiar with the Tinkercad environment (link3, tutorial video) • Students should have studied the educational material (slides) of Module_1-1, Module_1-2, Module_1-3, Module_1-4, and Module_1-5
Educational content	<p>Accompanying material:</p> <ul style="list-style-type: none"> • Module_1-5 slides • Module_1-5 Evaluation leaflet • Module_1-5 Open Projects

Tips

Tip. Keypad functions return characters, not numbers

Chapter 2: Activities

2.1 Activity 1. Basics with keypad 4x4

This activity uses a keypad 4x4. Five LEDs are connected to the Arduino Uno. The LEDs are turned on / off by the keypad.

Table 2. Activity 1

Activity 1 (50 minutes)	<ul style="list-style-type: none">• If "0" is pressed, all LEDs turn off• If "1" is pressed, the first LED turns on• If "2" is pressed, the second LED turns on• If "3" is pressed, the third LED turns on• If "4" is pressed, the fourth LED turns on• If "5" is pressed, the fifth LED turns on• If anything else is pressed, all LEDs turn on <p>Step 1. Draw the circuit in Tinkercad</p> <p>Step 2. Write the microcontroller code</p> <p>Step 3. Simulate the circuit and test it</p>
----------------------------	--

Step 1
(10 minutes)

Draw the next circuit in Tinkercad.

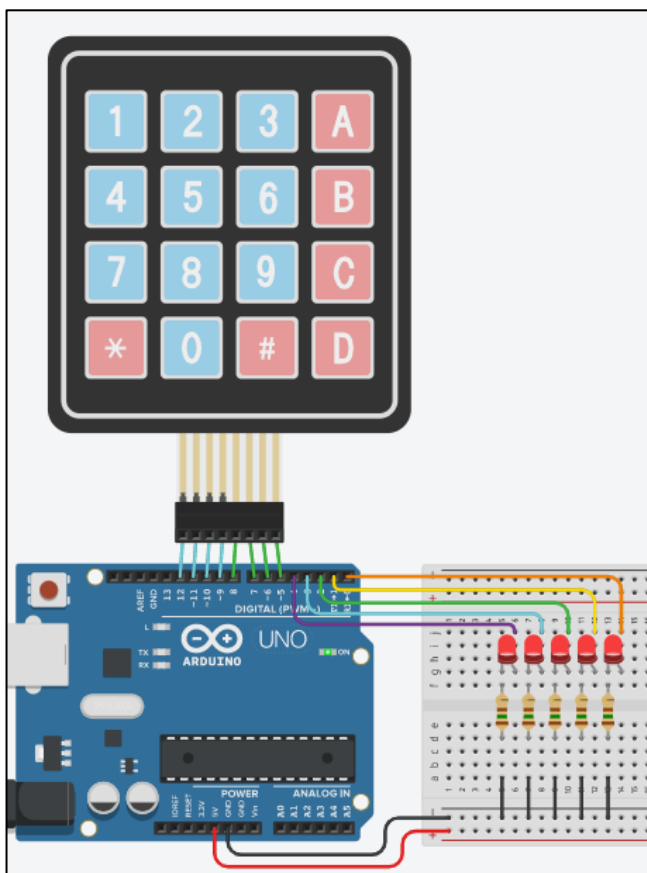


Figure 1. Keypad and LEDs

Step 2
(22 minutes)

Study the code and write it on the microcontroller:

```
/* Keypad and LEDs

Circuit Connections:
**LEDs
  PIN_0 => LED1_Anode - LED1_Cathode = > Resistor
150Ω => Gnd
  PIN_1 => LED2_Anode - LED2_Cathode = > Resistor
150Ω => Gnd
  PIN_2 => LED3_Anode - LED3_Cathode = > Resistor
150Ω => Gnd
  PIN_3 => LED4_Anode - LED4_Cathode = > Resistor
150Ω => Gnd
  PIN_4 => LED5_Anode - LED5_Cathode = > Resistor
150Ω => Gnd
**Keypad
  PIN_5 => Column4
  PIN_6 => Column3
  PIN_7 => Column2
  PIN_8 => Column1
  PIN_9 => Row4
  PIN_10 => Row3
  PIN_11 => Row2
  PIN_12 => Row1
*/

//include the library
#include <Keypad.h>

#define led1 0      //give the name "led1" to
PIN_0
#define led2 1      //give the name "led2" to
PIN_2
#define led3 2      //give the name "led3" to
PIN_2
#define led4 3      //give the name "led4" to
PIN_3
#define led5 4      //give the name "led5" to
PIN_4

const byte cols = 4; //four columns
const byte rows = 4; //four rows
//keypad output
char keys[rows][cols] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

//configure the Arduino Uno - Keypad interface
byte row_pins[] = {12,11,10,9}; //pins connect to
the rows
byte col_pins[] = {8, 7, 6, 5}; //pins connect to
the columns
Keypad keypad = Keypad(makeKeymap(keys), row_pins,
col_pins, rows, cols);
```

```

//variable to save keypad's characters
char key;

//The setup() function initializes and sets the
initial values
//It will only run once after each power up or
reset
void setup(){
  //Configure the PIN_0, PIN_1, PIN_2, PIN_3,
PIN_4
  //to behave as output
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
}

//loops consecutively
void loop(){
  //get a key from keypad
  key = keypad.getKey();
  //check if a key is pressed
  if (key != NO_KEY){
    //call the function "set_leds"
    set_leds(key);
  }
}

//this function configures the LEDs
//according to the key pressed
void set_leds(char x){
  if(x=='0'){ //all LEDs are OFF
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
    digitalWrite(led4, LOW);
    digitalWrite(led5, LOW);
  }
  else if(x=='1'){ //LED1 is ON
    digitalWrite(led1, HIGH);
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
    digitalWrite(led4, LOW);
    digitalWrite(led5, LOW);
  }
  else if(x=='2'){ //LED2 is ON
    digitalWrite(led1, LOW);
    digitalWrite(led2, HIGH);
    digitalWrite(led3, LOW);
    digitalWrite(led4, LOW);
    digitalWrite(led5, LOW);
  }
  else if(x=='3'){ //LED3 is ON
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
    digitalWrite(led3, HIGH);
    digitalWrite(led4, LOW);
    digitalWrite(led5, LOW);
  }
  else if(x=='4'){ //LED4 is ON
    digitalWrite(led1, LOW);

```

	<pre> digitalWrite(led2, LOW); digitalWrite(led3, LOW); digitalWrite(led4, HIGH); digitalWrite(led5, LOW); } else if(x=='5'){ //LED5 is ON digitalWrite(led1, LOW); digitalWrite(led2, LOW); digitalWrite(led3, LOW); digitalWrite(led4, LOW); digitalWrite(led5, HIGH); } else{ //all LEDs are ON digitalWrite(led1, HIGH); digitalWrite(led2, HIGH); digitalWrite(led3, HIGH); digitalWrite(led4, HIGH); digitalWrite(led5, HIGH); } } </pre>
<p>Step 3 (5 minutes)</p>	<p>Run the simulation and check the correct operation of the circuit</p>
<p>Step 4 (13 minutes)</p>	<p>Suggested modifications and discussion:</p> <ul style="list-style-type: none"> Adjust the code so that when “1” is pressed one LED turns on, when “2” is pressed two LEDs turn on ... when “5” is pressed five LEDs will turn on. Do we need to make any changes to the circuit?

2.2 Activity 2. Keypad 4x4 and LCD 16x2

In this activity the Arduino Uno reads a keypad. The key pressed appears on a liquid crystal display.

Table 3. Activity 2

<p>Activity 2 (50 minutes)</p>	<p>Step 1. Draw the circuit in Tinkercad Step 2. Write the microcontroller code Step 3. Simulate the circuit and test it Step 4. Modifications and discussion</p>
------------------------------------	--

Step 1
(13 minutes)

Draw the next circuit in Tinkercad.

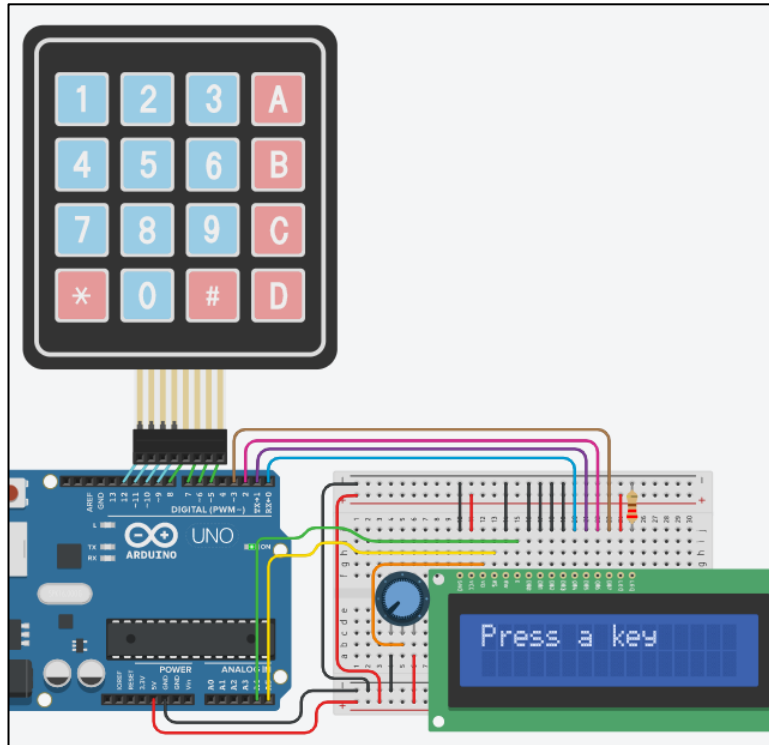


Figure 2. Keypad and LCD

Step 2
(17 minutes)

Study the code and write it on the microcontroller. The 2 missing lines must be completed:

```
/* Keypad and LCD

Circuit Connections:
** LCD
    Ground      => Gnd
    Power       => Vcc
    Contrast    => Potentiometer
    RS          => PIN_0
    RW          => Gnd
    E          => PIN_1
    DB0         => Gnd
    DB1         => Gnd
    DB2         => Gnd
    DB3         => Gnd
    DB4         => PIN_2
    DB5         => PIN_3
    DB6         => PIN_4
    DB7         => PIN_5
    LED Anode   => Vcc
    LED Cathode => Resistor 220Ω => Gnd
** Potentiometer1
    Terminal 1  => Gnd
    Wiper      => LCD_Contrast
    Terminal 2  => Vcc
**Keypad
    PIN_5      => Column4
    PIN_6      => Column3
    PIN_7      => Column2
    PIN_8      => Column1
    PIN_9      => Row4
    PIN_10     => Row3
    PIN_11     => Row2
    PIN_12     => Row1
*/

//include the library
#include <LiquidCrystal.h>
#define RS A5      //give the name "RS" to PIN_A5
#define EN A4      //give the name "EN" to PIN_A4
#define DB4 0      //give the name "DB4" to PIN_0
#define DB5 1      //give the name "DB5" to PIN_1
#define DB6 2      //give the name "DB6" to PIN_2
#define DB7 3      //give the name "DB7" to PIN_3

//configure the Arduino Uno - LCD interface
LiquidCrystal lcd(RS, EN, DB4, DB5, DB6, DB7);

//inlcude the library
#include <Keypad.h>

const byte cols = 4; //four columns
const byte rows = 4; //four rows

//keypad output
char keys[rows][cols] = {
    {'1', '2', '3', 'A'},
```

	<pre> {'4','5','6','B'}, {'7','8','9','C'}, {'*','0','#','D'} }; //configure the Arduino Uno - Keypad interface byte row_pins[] = {12,11,10,9}; //pins connect to the rows byte col_pins[] = {8, 7, 6, 5}; //pins connect to the columns Keypad keypad = Keypad(makeKeymap(keys), row_pins, col_pins, rows, cols); //variable to save keypad's characters char key; //The setup() function initializes and sets the initial values //It will only run once after each power up or reset void setup(){ //configure the LCD's columns and rows lcd.begin(16, 2); //print a message lcd.print("Press a key"); } //loops consecutively void loop(){ //wait until a key is pressed key=keypad.waitForKey(); //clear the LCD and print the key lcd.clear(); lcd.print(key); } </pre>
<p>Step 3 (5 minutes)</p>	<p>Run the simulation and check the correct operation of the circuit</p>
<p>Step 4 (15 minutes)</p>	<p>Suggested modifications and discussion:</p> <ul style="list-style-type: none"> • Could the application work instead of two with one push-button? • Add a switch. When the switch is open, the text on the LCD can be hidden by the corresponding push-button. When the switch is closed, the text on the LCD will be displayed whether a push-button is pressed or not. Write the appropriate code and run the simulation

2.3 Activity 3. Simple calculator

This activity uses a keypad and a LCD. The user can calculate the 4 basic mathematical operations between two single-digit positive numbers.

Table 4. Activity 3

<p>Activity 3 (50 minutes)</p>	<p>The calculator:</p> <ul style="list-style-type: none">• With the “A” key, makes the addition• With the “B” key, makes the subtraction• With the “C” key, makes the multiplication• With the “D” key, makes the division <p>The “#” key resets the result and a new calculation cycle begins</p> <p>Step 1. Draw the circuit in Tinkercad Step 2. Write the microcontroller code Step 3. Simulate the circuit and test it Step 4. Modifications and discussion</p>
------------------------------------	---

Step 1
(13 minutes)

Draw the next circuit in Tinkercad.

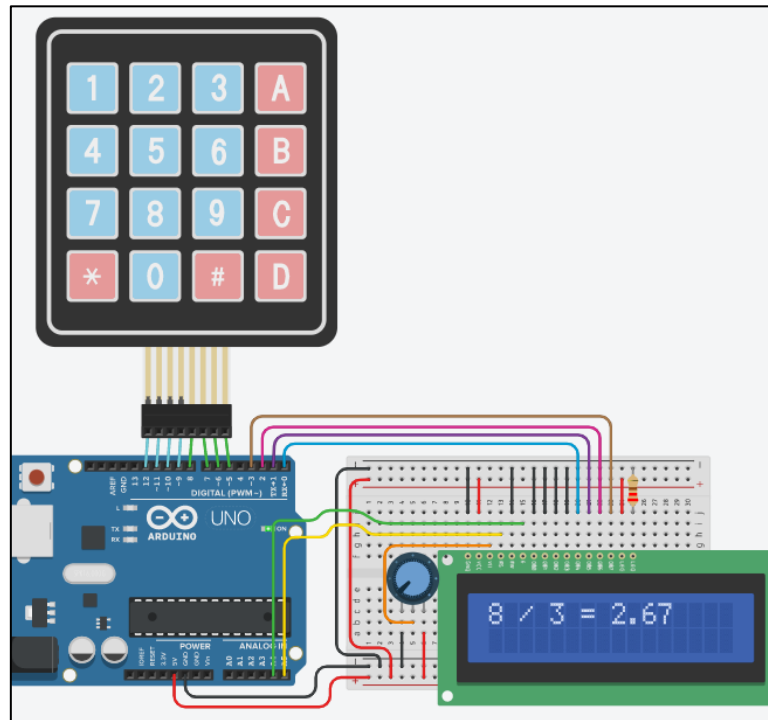


Figure 3. Simple calculator

Step 2
(25 minutes)

Study the code and write it on the microcontroller:

```
/* Simple Calculator

Circuit Connections:
** LCD
    Ground      => Gnd
    Power       => Vcc
    Contrast    => Potentiometer
    RS          => PIN_0
    RW          => Gnd
    E           => PIN_1
    DB0         => Gnd
    DB1         => Gnd
    DB2         => Gnd
    DB3         => Gnd
    DB4         => PIN_2
    DB5         => PIN_3
    DB6         => PIN_4
    DB7         => PIN_5
    LED Anode   => Vcc
    LED Cathode => Resistor 220Ω => Gnd
** Potentiometer1
    Terminal 1  => Gnd
    Wiper      => LCD_Contrast
    Terminal 2  => Vcc
**Keypad
    PIN_5      => Column4
    PIN_6      => Column3
    PIN_7      => Column2
    PIN_8      => Column1
    PIN_9      => Row4
    PIN_10     => Row3
    PIN_11     => Row2
    PIN_12     => Row1
*/

// include the library code:
#include <LiquidCrystal.h>
#define RS A5      //give the name "RS" to PIN_A5
#define EN A4      //give the name "EN" to PIN_A4
#define DB4 0      //give the name "DB4" to PIN_0
#define DB5 1      //give the name "DB5" to PIN_1
#define DB6 2      //give the name "DB6" to PIN_2
#define DB7 3      //give the name "DB7" to PIN_3

//configure the library with Arduino Uno - LCD
interface
LiquidCrystal lcd(RS, EN, DB4, DB5, DB6, DB7);

//include the library
#include <Keypad.h>
const byte cols = 4; //four columns
const byte rows = 4; //four rows

//keypad output
char keys[rows][cols] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
```

```

    {'7','8','9','C'},
    {'*','0','#','D'}
};

//configure the Arduino Uno - Keypad interface
byte row_pins[] = {12,11,10,9}; //pins connect to
the rows
byte col_pins[] = {8, 7, 6, 5}; //pins connect to
the columns
Keypad keypad = Keypad(makeKeymap(keys), row_pins,
col_pins, rows, cols);

//variable to save keypad's characters
char key;
//variable for the first number
int num1;
//variable for the second number
int num2;
//variable for the operation
char operation;
//variable for the result
float result;

//The setup() function initializes and sets the
initial values
//It will only run once after each power up or
reset
void setup() {
    //configure the LCD's columns and rows
    lcd.begin(16, 2);
}

//loops consecutively
void loop(){
    //the first number
    do{
        //wait until a key is pressed
        key=keypad.waitForKey();
        //call the "convert_to_number"
        num1=convert_to_number(key);
    }
    while(num1>9);
    //print the first number
    lcd.print(num1);
    //wait 0.2s
    delay(200);

    //the operation
    do{
        //wait until a key is pressed
        operation=keypad.waitForKey();
    }
    while(operation!='A'  &&  operation!='B'  &&
operation!='C' && operation!='D');

    //print the operation
    if(operation=='A'){
        lcd.print(" + ");
    }
    else if(operation=='B'){
        lcd.print(" - ");
    }
}

```

```

if(operation=='C'){
    lcd.print(" * ");
}
else if(operation=='D'){
    lcd.print(" / ");
}
//wait 0.2s
delay(200);

//the second number
do{
    //wait until a key is pressed
    key=keypad.waitForKey();
    //call the "convert_to_number"
    num2=convert_to_number(key);
}
while(num2>9);
//print the second number
lcd.print(num2);
//wait 0.2s
delay(200);

//calculate
if(operation=='A'){
    result=num1+num2;
}
else if(operation=='B'){
    result=num1-num2;
}
else if(operation=='C'){
    result=num1*num2;
}
else if(operation=='D'){
    result=float(num1)/num2;
}

//print the result
lcd.print(" = ");
lcd.print(result);

//wait until the "#" is pressed
do{
    key=keypad.waitForKey();
}
while(key!='#');

//clear the LCD
lcd.clear();
}

//this function converts
//keypad's character to integer
int convert_to_number(char c){
    if(c=='0'){
        return 0;
    }
    else if(c=='1'){
        return 1;
    }
    else if(c=='2'){
        return 2;
    }
}

```

	<pre> else if(c=='3'){ return 3; } else if(c=='4'){ return 4; } else if(c=='5'){ return 5; } else if(c=='6'){ return 6; } else if(c=='7'){ return 7; } else if(c=='8'){ return 8; } else if(c=='9'){ return 9; } else if(c=='A'){ return 10; } else if(c=='B'){ return 11; } else if(c=='C'){ return 12; } else if(c=='D'){ return 13; } else if(c=='*'){ return 14; } else { // if(c=='#'){ return 15; } } </pre>
<p>Step 3 (4 minutes)</p>	<p>Run the simulation and check the correct operation of the circuit</p>
<p>Step 4 (8 minutes)</p>	<p>Suggested modifications and discussion:</p> <ul style="list-style-type: none"> • Add an operation. Specifically, the power can be calculated between two single-digit positive numbers <p><i>Tip.</i> link</p>

Chapter 3: Recapitulation

The circuits were designed and simulated with Tinkercad.

Arduino Uno programming functions were used, such as:

- Keypad.waitForKey()
- Keypad.getKey()
- lcd.clear()

Through the activities were utilized

- Arduino Uno pins for driving a keypad 4x4

References

Breadboard \ Wiring. Retrieved from <http://wiring.org.co/learning/tutorials/breadboard/>

Brown, R. (2020). *Active vs. Passive buzzer: the differences*. Retrieved from <https://nerdytechy.com/active-vs-passive-buzzer/>

Learn C - Free Interactive C Tutorial. Retrieved from <https://www.learn-c.org/>

Learn how to use Tinkercad | Tinkercad. Retrieved from <https://www.tinkercad.com/learn/circuits>

