TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES

Ayala Boulevard, Ermita, Manila

CIT-ELECTRONICS DEPARTMENT

**CPET11L-M – Microprocessor and Microcontroller Systems, Lab**

**1st Semester, SY 2-24-2025**

|  |  |
| --- | --- |
| **Name:** Joseph C. Arenas | **Instructor:** Prof. Tristan Mopas |
| **Course & Section:** BET-CPET- 3A | **Date Submitted:**  September 20, 2025 |

**Activity 2**

**Topic 1: Arduino 7-Segment Display Counter Push Button**

**Topic 2: Segment Display Count Up Arduino**

**Topic 3: Arduino Push Button 2-Digit 7-Segment**

**Topic 4: Arduino Countdown Timer in Minutes & Seconds**

1. **OBJECTIVES**

* To apply practical knowledge in using the Arduino Uno R3 and Mega
* To explain the functions and components of the related topics
* To implement the 7-Segment Display, Dual Digit 7-Segment Display, and 4-Digit 7-Segment Display using code and circuit diagrams
* To develop and enhance problem-solving skills related to the topics

1. **EQUIPMENT AND MATERIALS**

**HARDWARE**

* Arduino Uno or Arduino Mega 2560
* Breadboard
* Jumper Wires
* Push Button
* Laptop
* 7-Segment Display & 4-Digit 7-Segment Display
* Resistors 220Ω and 10KΩ
* Buzzer

**SOFTWARE**

* Arduino IDE
* MS Word
* TinkerCad Simulator
* Wokwi Simulator

1. **DIAGRAM**

**===== Topic 1: Arduino 7-Segment Display Counter Push Button =====**

1. **TinkerCad Simulation**

**A circuit board with wires

AI-generated content may be incorrect.**

1. **Breadboard**

**A digital display with wires

AI-generated content may be incorrect.**

1. **Source Code**

if(digit !=2)

digitalWrite(c,HIGH);

if(digit != 1 && digit !=4 && digit !=7)

digitalWrite(d,HIGH);

if(digit == 2 || digit ==6 || digit == 8 || digit==0)

digitalWrite(e,HIGH);

if(digit != 1 && digit !=2 && digit!=3 && digit !=7)

digitalWrite(f,HIGH);

if (digit!=0 && digit!=1 && digit !=7)

digitalWrite(g,HIGH);

}

void turnOff(){

digitalWrite(a,LOW);

digitalWrite(b,LOW);

digitalWrite(c,LOW);

digitalWrite(d,LOW);

digitalWrite(e,LOW);

digitalWrite(f,LOW);

digitalWrite(g,LOW);

}

const int a = 8;

const int b = 9;

const int c = 4;

const int d = 5;

const int e = 6;

const int f = 2;

const int g = 3;

bool bPress = false;

const int buttonPin = 10;

int buttonPushCounter = 0;

int buttonState = 0;

int lastButtonState = 0;

void setup() {

pinMode(a, OUTPUT); //A

pinMode(b, OUTPUT); //B

pinMode(c, OUTPUT); //C

pinMode(d, OUTPUT); //D

pinMode(e, OUTPUT); //E

pinMode(f, OUTPUT); //F

pinMode(g, OUTPUT); //G

pinMode( buttonPin , INPUT\_PULLUP );

Serial.begin(9600);

displayDigit(buttonPushCounter);

}

void displayDigit(int digit){

if(digit!=1 && digit != 4)

digitalWrite(a,HIGH);

if(digit != 5 && digit != 6)

digitalWrite(b,HIGH);

**===== Topic 2: Segment Display Count Up Arduino =====**

1. **TinkerCad Simulation**

**A circuit board with wires

AI-generated content may be incorrect.**

1. **Breadboard**

**A circuit board with wires and a box with wires

AI-generated content may be incorrect.**

1. **Source Code**

byte pin[] = {2, 3, 4, 5, 6, 7, 8, 9};

int number[9][8] = {

{1, 1, 0, 0, 0, 1, 1, 1},

{0, 0, 1, 0, 0, 0, 1, 0},

{1, 0, 0, 0, 0, 0, 1, 0},

{1, 1, 0, 0, 0, 1, 0, 0},

{1, 0, 0, 0, 1, 0, 0, 0},

{0, 0, 0, 0, 1, 0, 0, 0},

{1, 1, 0, 0, 0, 0, 0, 1},

{0, 0, 0, 0, 0, 0, 0, 0},

{1, 1, 0, 0, 0, 0, 0, 0},

};

void setup() {

for (byte a = 0; a < 8; a++) {

pinMode(pin[a], OUTPUT);

}

}

void loop() {

for (int a = 0; a < 9; a++) {

for (int b = 0; b < 8; b++) {

digitalWrite(pin[b], number[a][b]);

}

delay(500);

}

}

**=== TOPIC 3: Arduino Push Button 2-Digit 7-Segment ===**

1. **Wokwi Simulation**

**A circuit board with wires and a digital display

AI-generated content may be incorrect.**

1. **Breadboard**

**A circuit board with wires and a red light

AI-generated content may be incorrect.**

1. **Source Code**

int leds\_Uno[] = {2, 3, 4, 5, 6, 7, 8};

int leds\_Dos[] = {22, 23, 24, 25, 26, 27, 28};

int button\_Uno = 9;

int button\_Dos = 10;

int numbers = 0;

int digit\_Array[10][7] = {

{1, 1, 1, 1, 1, 1, 0},

{0, 1, 1, 0, 0, 0, 0},

{1, 1, 0, 1, 1, 0, 1},

{1, 1, 1, 1, 0, 0, 1},

{0, 1, 1, 0, 0, 1, 1},

{1, 0, 1, 1, 0, 1, 1},

{1, 0, 1, 1, 1, 1, 1},

{1, 1, 1, 0, 0, 0, 0},

{1, 1, 1, 1, 1, 1, 1},

{1, 1, 1, 1, 0, 1, 1},

};

void setup(){

for(int i = 0 ; i < 7 ; i++){

pinMode(leds\_Uno[i], OUTPUT);

pinMode(leds\_Dos[i], OUTPUT);

}

pinMode(button\_Uno, INPUT);

pinMode(button\_Dos, INPUT);

}

void segmentOutput(int pins[], int digit){

for(int i = 0 ; i < 7 ; i++){

digitalWrite(pins[i], digit\_Array[digit][i]);

}

}

void loop(){

if (digitalRead(button\_Uno) == HIGH){

numbers--;

if (numbers < 0) numbers = 99;

delay(150);

}

if (digitalRead(button\_Dos) == HIGH){

numbers++;

if (numbers > 99) numbers = 0;

delay(150);

}

int tens = numbers / 10;

int ones = numbers % 10;

segmentOutput(leds\_Uno, tens);

segmentOutput(leds\_Dos, ones);

}

**=== Topic 4: Arduino Countdown Timer in Minutes & Seconds ===**

1. **Wokwi Simulation**

**A circuit board with wires connected to it

AI-generated content may be incorrect.**

1. **Breadboard**

**A circuit board with wires

AI-generated content may be incorrect.**

1. **Source Code**

#include <math.h>

int digit\_pin[] = {6, 9, 10, 11}; // PWM Display digit pins from left to right

int speakerPin = 15;

#define DIGIT\_ON LOW

#define DIGIT\_OFF HIGH

int segA = 2;

int segB = 3;

int segC = 4;

int segD = 5;

int segE = A0; //pin 6 is used bij display 1 for its pwm function

int segF = 7;

int segG = 8;

//int segPD = ;

int button1=13;

int button2=12;

int button3=16;

int button4=17;

int countdown\_time = 60;

struct struct\_digits {

int digit[4];

};

void setup() {

pinMode(segA, OUTPUT);

pinMode(segB, OUTPUT);

pinMode(segC, OUTPUT);

pinMode(segD, OUTPUT);

pinMode(segE, OUTPUT);

pinMode(segF, OUTPUT);

pinMode(segG, OUTPUT);

for (int i=0; i<4; i++) {

pinMode(digit\_pin[i], OUTPUT);

}

pinMode(speakerPin, OUTPUT);

pinMode(button1,INPUT\_PULLUP);

pinMode(button2,INPUT\_PULLUP);

pinMode(button3,INPUT\_PULLUP);

pinMode(button4,INPUT\_PULLUP);

}

void playTone(int tone, int duration) {

for (long k = 0; k < duration \* 1000L; k += tone \* 2) {

digitalWrite(speakerPin, HIGH);

delayMicroseconds(tone);

digitalWrite(speakerPin, LOW);

delayMicroseconds(tone);

}

}

case 3:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_ON);

break;

case 4:

digitalWrite(segA, SEGMENT\_OFF);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_OFF);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

case 5:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_OFF);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

void lightNumber(int numberToDisplay) {

#define SEGMENT\_ON HIGH

#define SEGMENT\_OFF LOW

switch (numberToDisplay){

case 0:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_ON);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_OFF);

break;

case 1:

digitalWrite(segA, SEGMENT\_OFF);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_OFF);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_OFF);

break;

case 2:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_OFF);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_ON);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_ON);

break;

case 9:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

case 10:

digitalWrite(segA, SEGMENT\_OFF);

digitalWrite(segB, SEGMENT\_OFF);

digitalWrite(segC, SEGMENT\_OFF);

digitalWrite(segD, SEGMENT\_OFF);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_OFF);

break;

}

}

void SwitchDigit(int digit) {

for (int i=0; i<4; i++) {

if (i == digit) {

digitalWrite(digit\_pin[i], DIGIT\_ON);

} else {

digitalWrite(digit\_pin[i], DIGIT\_OFF);

}

}

}

case 6:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_OFF);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_ON);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

case 7:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_OFF);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_OFF);

break;

case 8:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_ON);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

bool Countdown(int n, int del){

int minutes = n / 100;

int seconds = n % 100;

while (minutes > 0 || seconds > 0) {

// Combine back into MMSS format for display

int displayTime = minutes \* 100 + seconds;

PrintNumber(displayTime, del);

if (digitalRead(button2) == LOW) {

return false; // stop pressed

}

// countdown logic like a real clock

if (seconds == 0) {

if (minutes > 0) {

minutes--;

seconds = 59;

}

} else {

seconds--;

}

}

// Final 00:00

PrintNumber(0,0);

playTone(1519,1000);

return true;

}

struct struct\_digits IntToDigits(int n){

struct struct\_digits dig;

int zeros=0;

int d;

for (int i=0; i<4; i++) {

d=n/pow(10,3-i);

zeros += d;

n = n - d\*pow(10,3-i);

if (zeros!=0 || i==3) {

dig.digit[i]=d;

} else {

dig.digit[i]=10;

}

}

return dig;

}

void PrintNumber(int n, int time) {

struct struct\_digits dig;

dig = IntToDigits(n);

for (int i=0; i<= time/20; i++) {

if (digitalRead(button2)==LOW) {

return;

}

for (int j=0; j<4; j++) {

SwitchDigit(j);

lightNumber(dig.digit[j]);

delay(5);

}

}

}

else if (digitalRead(button4)==LOW) {

if (pressed4 == 0 || pressed4 > 30) {

int minutes = countdown\_time / 100;

int seconds = countdown\_time % 100;

// cap at 99:59 to stay within 4 digits

if (!(minutes == 99 && seconds == 59)) {

if (seconds == 59) {

if (minutes < 99) { minutes++; seconds = 0; }

} else {

seconds++;

}

countdown\_time = minutes \* 100 + seconds;

dig = IntToDigits(countdown\_time);

}

}

pressed4 += 1;

}

if (digitalRead(button3)==HIGH) pressed3 = 0;

if (digitalRead(button4)==HIGH) pressed4 = 0;

}

}

void reset() {

int m, zeros, d, pressed3 = 0, pressed4 = 0;

countdown\_time = 1200;

struct struct\_digits dig;

dig = IntToDigits(countdown\_time);

while (digitalRead(button1)==HIGH) {

for (int j=0; j<4; j++) {

SwitchDigit(j);

lightNumber(dig.digit[j]);

delay(5);

}

// Decrement (one second, borrow to minutes when needed)

if (digitalRead(button3)==LOW) {

if (pressed3 == 0 || pressed3 > 30) {

int minutes = countdown\_time / 100;

int seconds = countdown\_time % 100;

if (minutes > 0 || seconds > 0) {

if (seconds == 0) {

if (minutes > 0) { minutes--; seconds = 59; }

} else {

seconds--;

}

countdown\_time = minutes \* 100 + seconds;

dig = IntToDigits(countdown\_time);

}

}

pressed3 += 1;

}

void loop(){

reset();

while (!Countdown(countdown\_time,962)) {

reset();

}

while (digitalRead(button2)==1){};

}

1. **PROCEDURE**

**A. Preparation**

* Gather the required components as stated in Chapter 2
* Prepare the Arduino IDE on your preferred device.
* Review the problem for each topic and formulate or search for a circuit diagram given the problem of each topic.
* Simulate your circuit diagram to a circuit simulator such as TinkerCADor Wokwi.

**B. Actual**

* For topic 1, connect each segment of the 7 segment display to their respective digital pins as per the circuit diagram. Connect the common segments of the 7-segment display to ground as it is a common anode configurated component. Attach the tactile/push button on the breadboard and connect digital pin 10 to it while connecting its adjacent pin to VCC. Connect the common segments of the 7-segment display to ground as it is a common anode configurated component using a 220Ω resistor. Connect the 5v VCC and GND pin of the microcontroller to the “+” and “-“ of the breadboard. Connect the USBVCC to the Arduino Uno R3 / Arduino Mega 2560 then verify and upload the code.
* For topic 2, connect each segment of the 7 segment display to their respective digital pins as per the circuit diagram. Connect the common segments of the 7-segment display to ground as it is a common anode configurated component using a 220Ω resistor. Connect the 5v VCC and GND pin of the microcontroller to the “+” and “-“ of the breadboard. Connect the USBVCC to the Arduino Uno R3 / Arduino Mega 2560 then verify and upload the code.
* For topic 3, connect the first 7-segment display’s segments to digital pins 2-8. Connect the second 7-segment display’s segments to digital pins 22-28. Connect the common segments of all the 7-segment displays to ground as it is a common anode configurated component using a 220Ω resistor. Place the two push buttons on the bread board. Connect digital pin 9 on the first button then direct it to ground for a normally low, active high button. Connect the adjacent pin of the first button to VCC. Repeat for the second button but connect it to digital pin 10. Connect the 5v VCC and GND pin of the microcontroller to the “+” and “-“ of the breadboard. Connect the USBVCC to the Arduino Uno R3 / Arduino Mega 2560 then verify and upload the code.
* For topic 4, connect the segments of the 4-digit 7-segment display to their respective digital pins. Connect the digit segments to digital pins 6, 9, 10, and 11 with a 220Ω resistor. Attach four push buttons on the breadboard with their pins connected to 13, 12, 16 and 17 respectively. Connect each adjacent pin of the 4 push buttons to VCC. Attach the buzzer to the breadboard. Connect its anode to digital pin 15. Connect its cathode to ground. Connect the 5v VCC and GND pin of the microcontroller to the “+” and “-“ of the breadboard. Connect the USBVCC to the Arduino Uno R3 / Arduino Mega 2560 then verify and upload the code.

**C. Checking**

* For topic 1, ensure that at each press of the button, 7-segment display number increments by 1 from numbers 0-9 then back to 0.
* For topic 2, ensure that the 7-segment display shows the numbers 0-9 then back to 0 incrementing automatically.
* For topic 3, ensure that numbers of 1 digit and two digits increment when button 2 is pressed and decrements when button 1 is pressed. Numbers displayed should be from 0-99.
* For topic 4, ensure that proper time formatting is followed. Ensure that button 1 starts the timer, button 2 resets the timer back to 12:00, button 3 decrements the timer alarm and button 4 increments the timer alarm. Make sure that when the timer reaches 00:00, the buzzer goes HIGH.
* Adjust values or fix errors if the output is not working as expected.

**D. Uploading**

* Upload the final and corrected code to the Arduino Uno R3/Arduino Mega 2560
* Ensure that all four topics properly work.

1. **CONCLUSION**

The activities highlighted the practical integration of hardware and software using Arduino, particularly with 7-segment displays and LEDs. They demonstrated how digital logic, coding, and circuit design work together to produce functional outputs. Overall, the tasks strengthened both technical understanding and problem-solving skills in microcontroller applications.

A close-up of a document

AI-generated content may be incorrect.