ENGR 101: Exam

2023, June 12 ** WITH SOLUTIONS **

Instructions

- Time allowed: **TWO HOURS**
- Attempt **all** the questions. There are **70 marks** in total.
- Brief Documentation is at the end of the examination script
- Write your answers in this exam paper and hand in all sheets.
- If you think some question is unclear, ask for clarification.
- You may use unmarked paper Chinese-English translation dictionaries.
- You may write notes and working on this paper, but make sure your answers are clear.

| Questions	Marks	
1. Finite State machines (FSM)	[28]	
2. Arduino programming	[20]	
3. Wiring up an Arduino	[12]	
4. FSM and Arduino	[10]	
	TOTAL:	

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 1. FSM controllers   **[28 marks]**

The diagram below shows a FSM controller robotic toy. You have seen this FSM controller in the lab.
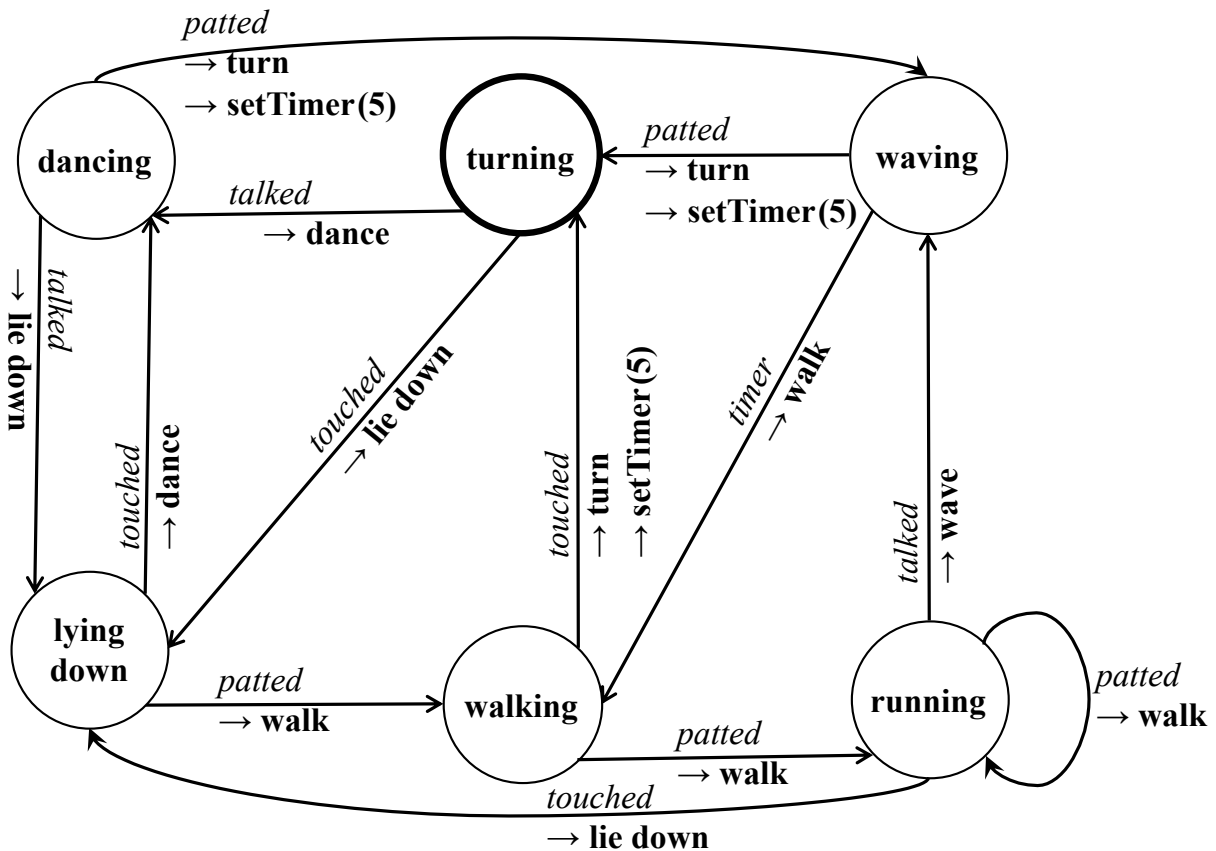
It has the following sensors and actions:

Sensors	Actions
patted	→ walk
touched	→ dance
talked	→ wave
timer	→ lie down
	→ turn
	→ setTimer(5)

(a) **[6 marks]** If the controller starts in the state **turning** and gets this sequence of sensor values, what sequence of **states** will the controller go through?

Sensor sequence: *talked*, *patted*, *timer*, *patted*, *touched*, *touched*

State: dancing, waving, walking
running, lying down, dancing

(Question 1 continued)

(b) **[12 marks]** If the controller starts in the state **walking** and gets this sequence of sensor values, what sequence of **actions** will the controller do, and what state will it end in?

Sensor sequence: *patted, talked, patted, talked, talked, touched , patted*

Actions: walk, wave, trun, setTimer(5), dance, lie down, dance, trun,
 setTimer(5)

Final State: waving

(c) **[10 marks]** Suppose the machine is in the state **running**. Give two different sequences of sensors that would make the machine go to the state **dancing**.

Sequence 1: talked, patted, talked

Sequence 2: touched, touched

Question 2. C programming [20 marks]

(a) **[8 marks]** What will be the output of the following Arduino sketch?

```
void setup(){
    Serial.begin(9600);
    int num = 8;
    for ( int i = 1; i <= num; i++) {
        if (i % 3 == 0) {
            Serial.print("Hop");
        } else {
            Serial.print(i);
        }
    }
}
void loop(){
}
```

Hint: *write the values of the variables in the boxes*

num: [   ]

i: [   ]

12Hop45Hop78

(b) **[4 marks]** Write an Arduino sketch to divide the numbers in variables num1 and num2 and display the result on the Serial Monitor?

```
double num1 = 2.5;
double num2 = 6.0;
void setup(){


    Serial.begin(9600);
    double ans = num1 / num2;
    Serial.println(ans);


}
void loop(){
}
```

(Question 2 continued)

(c) **[8 marks]** Write an Arduino sketch to display all the numbers in the given array on the Serial Monitor? You must use a loop (for or while).

```
const int length = 12;
int arr[length] = { −4, 0, 2, 3, 7, 12, 8, 9, −10, 3, 1, 23 };
void setup() {
    Serial.begin(9600);

  int i = 0;
  while (i < length) {
      Serial.println(arr[i]);
      i++;
  }
   // OR

    for (int i = 0; i < length; i++) {
        Serial.println(arr[i]);
    }


}
void loop(){
}
```

Question 3. Programming and Wiring up an Arduino [12 marks]

A photoresistor can be connected to an Arduino pin to turn an LED on or off.

(a) **[5 marks]** Use the provided components in the picture below to draw a circuit for an Arduino project that uses a photoresistor as an input and an LED as an output.

The circuit should turn on the LED when Pin 8 is set to HIGH, and the photoresistor should send varying voltages based on the light density to Pin A5.
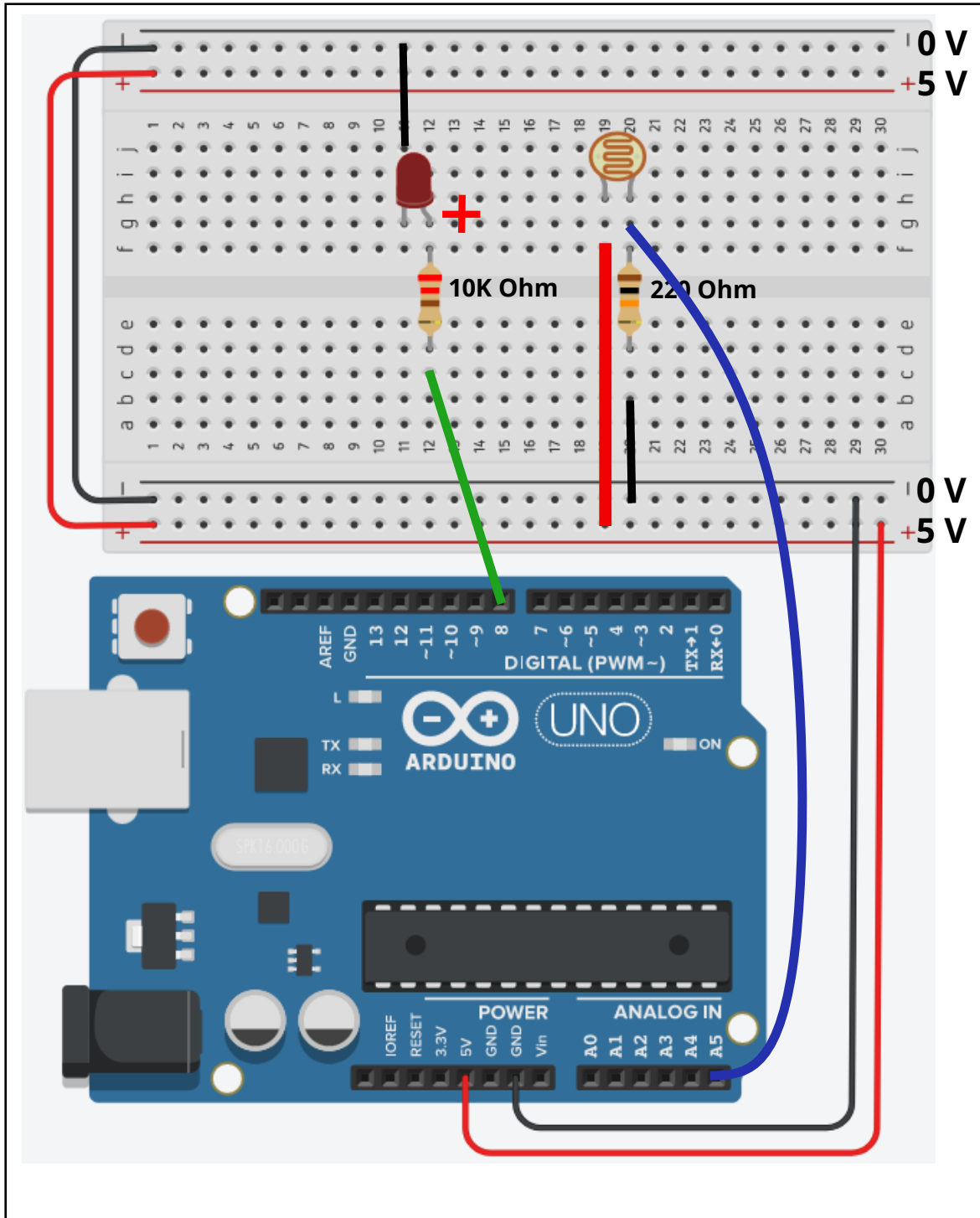
(Question 3 continued)

(b) **[7 marks]** Write an Arduino sketch that turns on an LED when the value reading from a photoresistor is greater than or equal to a specified threshold, and turns off the LED when the value reading is less than the threshold.

Note: The photoresistor is connected to analog pin A5 and the LED is connected to digital pin 8. The threshold value is set to 500.

```cpp
const int lightPin = A5;          // define photoresistor pin
const int ledPin = 8;             // define LED pin
const int threshold = 500;        // Set darkness threshold
void setup(){
    // Set up the pins

    pinMode(ledPin, OUTPUT);      // LED pin as output
    pinMode(lightPin, INPUT);     // photoresistor pin as input


}
void loop(){
    // Read the pin and perform actions



    int val = analogRead(lightPin); // Read light value

    // Turn LED on if light value is below threshold
    if ( val >= threshold ) {
        digitalWrite (ledPin, HIGH);
    } else {
        digitalWrite (ledPin, LOW);
    }






}
```
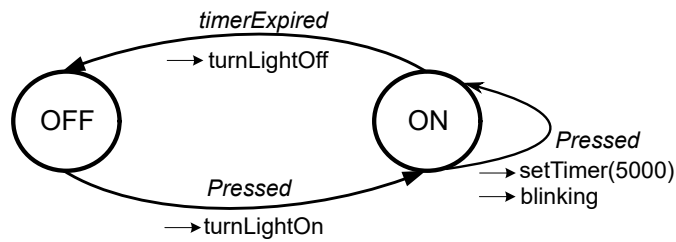
Question 4. FSM and Arduino [10 marks]

Consider the following event-driven Finite State Machine controller for a simple Arduino device.

The Arduino device has two states and three transitions

- The device sends the signal Pressed and timerExpired to the controller.
- The controller can perform four different actions on the device: turnLightOn(), turnLightOff(), blinking(), and setTimer(n) where n is milliseconds.

Part of the code for the controller is on the facing page:

- The state variable contains the current state of the device.
- The pressed() function returns HIGH if a user pushes the button,
- The checkTimer() function checks if a timer has expired. If the timer has expired, it sets timerExpired variable to True.

(Question 4 continued)

```
String state   = "OFF"; // Stores the current state of the FSM
bool timerExpired = false;

void setup(){
    pinMode(Light, OUTPUT); // Set pin as an output
}

void loop(){
    checkTimer();
    int pressed = pressed(); // returns HIGH if the user has pushed the button

    // actions and transitions
    if (state ==    "OFF"          ){
        if ( pressed == HIGH        ){

            turnLightOn();
            state = "ON";

        }
    } else if (state ==     "ON"            ){
        if ( pressed == HIGH          ){
            setTimer(5000);
            blinking ();
            state = "ON";

        }else if (    timerExpired == True    ){

            turnLightOff ();
            state = "OFF";

        }
    }

    if (timerExpired = true){ // Reset the timer
        timerExpired = false;
    }
}
/** Check if timer expired */
void checkTimer() {
    if ( millis () − timerStart > timerEnd) {
        timerEnd = 0;
        timerExpired = true;   // tell the controller that the timer ran out.
    }
}
```

* * * * * * * * * * * * * *

Documentation

Brief, simplified specifications of some relevant Arduino functions.

Serial

Serial . *begin*(speed)   // *opens serial port , sets data rate to speed bps*
Serial . *print* (anything val)   // *Prints val with no newline*
Serial . *println* (anything val)   // *Prints val and newline*
Serial . *println* ()   // *Prints a newline*

Digital I/O

void *pinMode*(*int* pin , *int* mode)   // *Configures the   specified   pin as INPUT or OUTPUT*
void *digitalWrite* (*int* pin , *int* value) // *Write a HIGH or a LOW value to a digital pin .*
int *digitalRead* (*int* pin)   // *read HIGH or LOW from the input pin*

Analog I/O

int analogRead(*int* pin)   // *Reads the value from the specified analog pin .*
                                                                   // *The return value is between 0 and 1023.*
void *analogWrite*(*int* pin ,   *int* value)   // *Writes an analog value (PWM wave) to a pin.*

Time

void *delay*(unsigned *long* ms)   // *Pauses the program for the amount of time (in   milliseconds)*
void *delayMicroseconds*(unsigned *int* us) // *Pauses the program for the amount of time (in microseconds)*

Random Numbers

long random(*long* min, *long*   max)   // *The random function generates pseudo−random numbers.*
void *randomSeed*(unsigned *long*)   //   *Initializes   the pseudo−random number generator*

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.