CONTENTS

CONTENTS

Date : 12/08/2024 Expt. No. : 1 Page No. : 1

Introduction to MSP430 Launch Pad and Programming Environment

Aim :

To familarize with MSP430 Launch Pad microcontroller, its programming environment

Components :-

- Arduino
- Bread board
- Wire

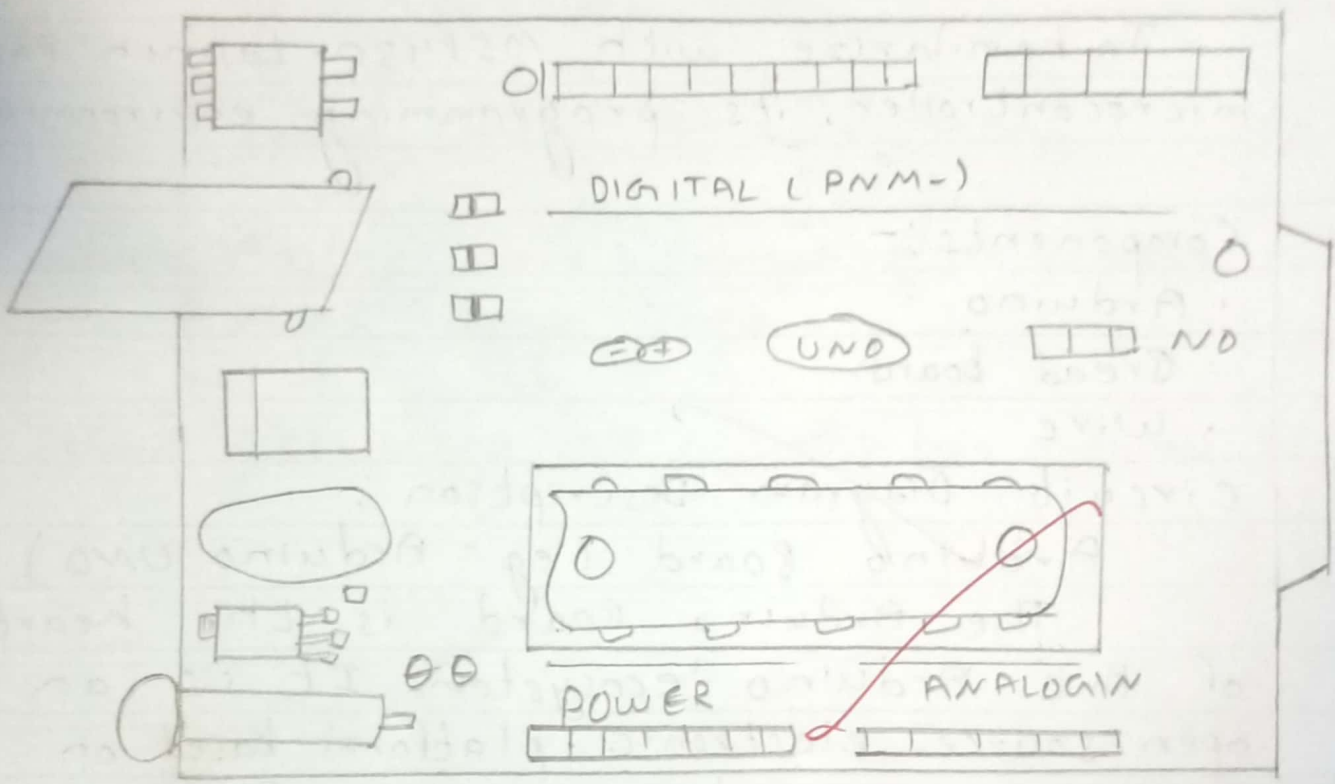
Circuit Diagram Description :

Arduino Board (eg : Arduino UNO)

The Arduino Board is the heart of the Arduino ecosystem. It is an open-source electronic platform based on easy-to-use hardware and software. The Arduino UNO is one of the most popular models and is based on the ATmega328 microcontroller.

-> Microcontroller : ATmega328P
-> operating voltage : 5V
-> Digital I/O pins : 14
-> Analog input pins : 6
-> flash memory : 32 bit
-> clock speed : 16MHz

Circuit Diagram:-



-> Connectivity: USB port for programming and power, DC Power Jack for external power supply.

Program :-

a)
```
void setup() {
  Serial.begin(115200);
  Serial.println("Hello, ESP32!");
}
void loop() {
  delay(10);
}
```

b)
```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILIN, LOW);
  delay(1000);
}
```

c)
```
int LEDB = 12;
void setup() {
pinmode( LEDB OUTPUT);
}
void loop() {
digitalwrite (LEDB, HIGH);
delay(1000);
digitalwrite ( LEDB, LOW);
delay (1000);
}
```

d)
```
Void setup(){
pinMode (12, OUTPUT);
}
void loop(){
digital write(12, HIGH);
delay(1000);
digital write (12, LOW);
delay(1000);
}
```

Result:- MSP430 launch pad and it's Programming & environment is successfully familarized.

21/8/21

Date : 29/08/2024 Expt. No. : 2 Page No. : 4

Aim: To read input from switch and automatic control / flash LED (software delay).

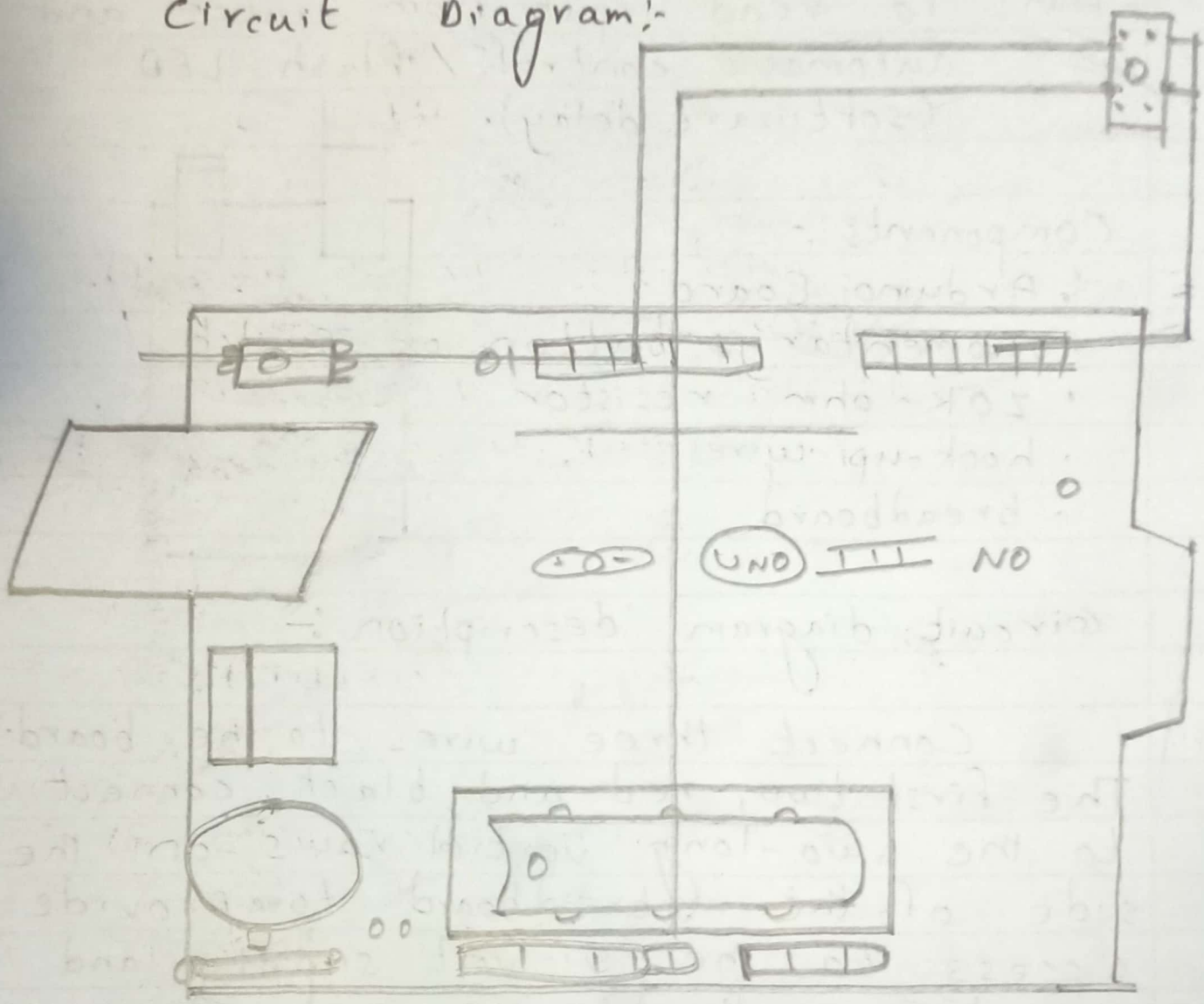
Components :-

- Arduino Board
- Momentary button or switch
- 10K ohm resistor
- hook-up wires
- breadboard

Circuit diagram description :-

Connect three wires to the board. The first two, red and black, connect to the two long vertical rows on the side of the breadboard to provide access to the 5 volt supply and ground. The third wire goes from digital pin 2 to one leg of the pushbutton. That same leg of the button connects through a pull down resistor (here 10k ohm) to ground. The other leg of the button connects to the 5 volt supply.

Circuit Diagram:-



Date : Expt. No. : Page No. : 5

- When the pushbutton is open there is no connection between the two legs of the pushbutton, so the pin is connected to ground and we read a LOW. When the button is closed, it makes a connection between its two legs, connecting the pin to 5 volts so that we read a HIGH.

You can also wire this circuit the opposite way, with a pullup resistor keeping the input HIGH, and going LOW when the button is pressed. If so, the behaviour of the sketch will be reversed, with the LED normally on and turning off when you press the button.

If you disconnect the digital I/O pin from everything, the LED may blink erratically. This is because the input is "floating" - that is, it will randomly return either HIGH or LOW. That's why you need a pull-up or pull-down resistor in the circuit.

Program :-

```
//constants won't change.
const int buttonPin = 2;
const int ledPin = 13;
int buttonstate = 0;

void setup(){
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
  }

void loop(){
buttonstate = digitalRead(button Pin);
if(buttonstate == HIGH){
 digitalWrite(ledPin, HIGH);
  delay(20000);
 } else {
  digitalWrite(ledpin, LOW);
   delay(20000);
 }
}
```

Result: Input from switch and automatic control / flash LED is successfully read.

21/8/22

Date : 21/08/2024 Expt. No. : 3 Page No. : 7

Aim:- To interrupts programming example using GPIO.

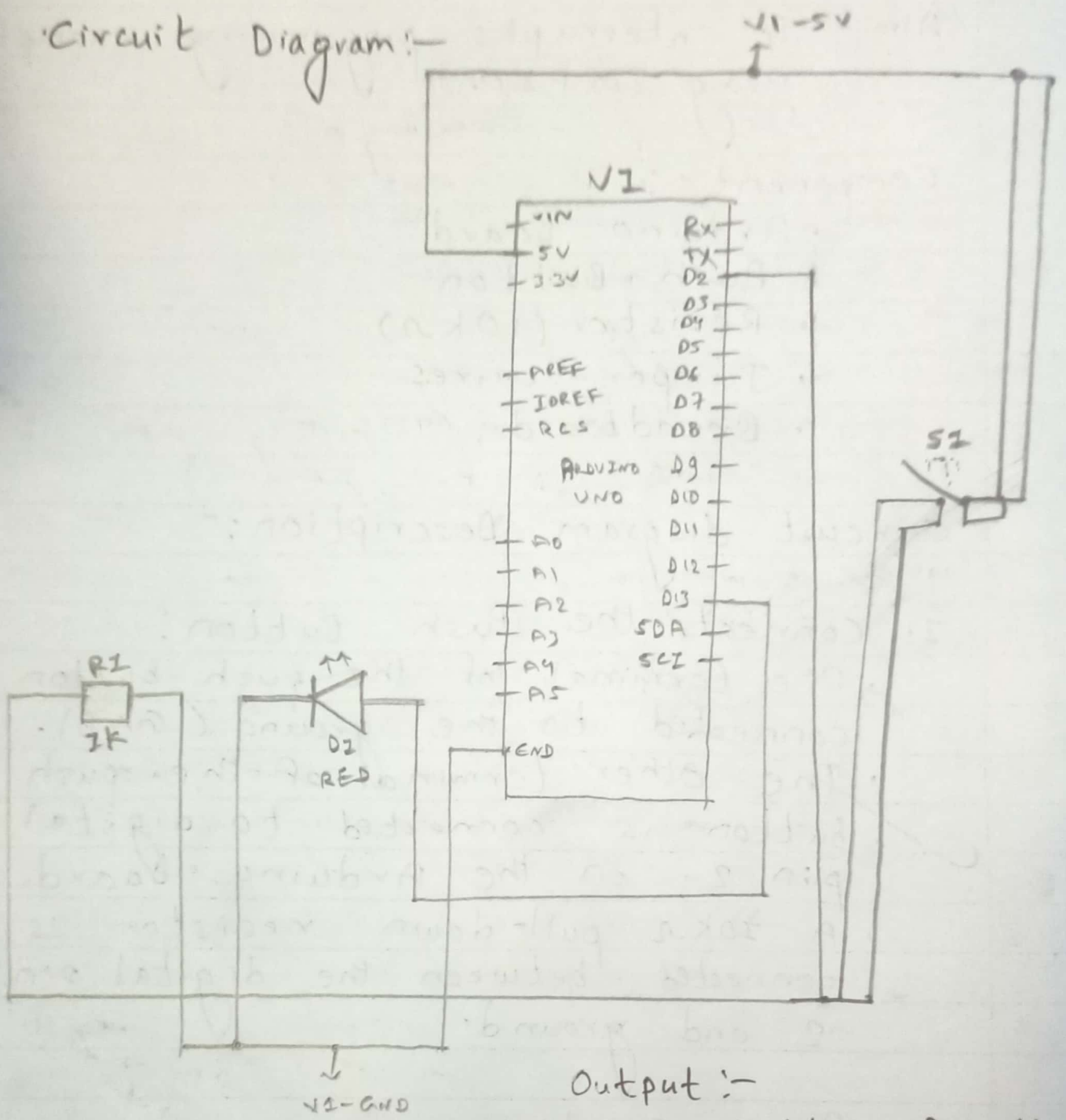
Components:
- Arduino Board
- Push Button
- Resistor (10kΩ)
- Jumper Wires
- Breadboard

Circuit diagram Description:-

1. Connect the Push Button:
   - One terminal of the push button is connected to the ground (GND).
   - The other terminal of the push button is connected to digital pin 2 on the Arduino board
   - A 10kΩ pull-down resistor is connected between the digital pin 2 and ground.

2. Power supply:
   - Connect the 5V pin of the Arduino board can be to the power rail on the breadboard.

Circuit Diagram :-



Output :-

Button Pressed!

Button Pressed!

Date : Expt. No. : Page No. : 8

Program :-

```
int pin = 2;
int checkPin;
void set() {
Serial.begin(9600);
int checkPin = digitalPinToInterrupt(pin
if(checkPin == -1){
  serial.println("Not a valid interruptpin!");
} else{
 Serial.println("valid interrupt pin");
}}
void loop()
  {
   }
```

Result :-

Hence, the interrupts programming example using GPIO.

26/8/24

Date : 28/08/2024 Expt. No. : 4 Page No. : 9

Aim :- To configure watchdog timer in watchdog and interval mode.

Components :-

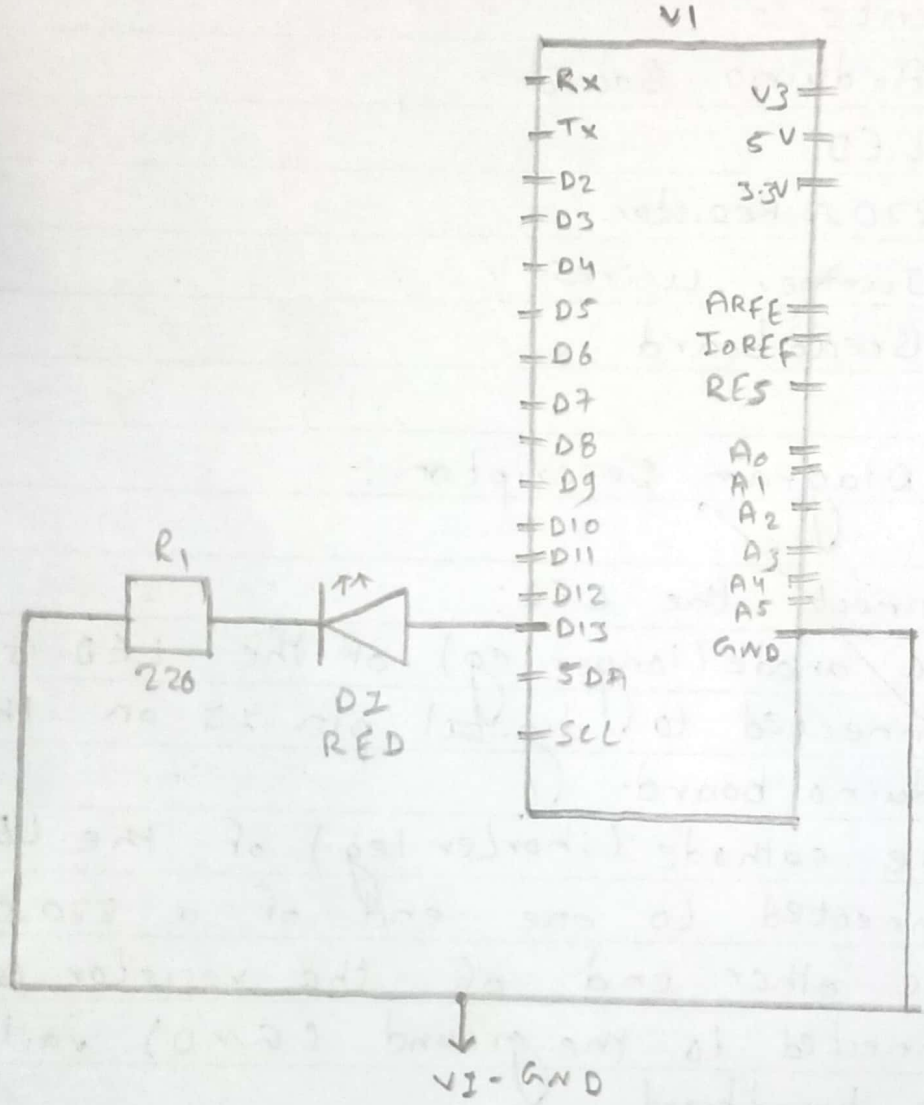
- Arduino Board
- LED
- 220Ω Resistor
- Jumper Wires
- Breadboard

Circuit Diagram Description :-

1. Connect the LED:
   - The anode (longer leg) of the LED is connected to digital pin 13 on the Arduino board.
   - The cathode (shorter leg) of the LED is connected to one end of a 220Ω resistor.
   - The other end of the resistor is connected to the ground (GND) rail on the breadboard.

2. Power Supply:
   - Connect the 5V pin of the Arduino to the power rail on the breadboard.
   - Connect the GND pin of the Arduino to the ground rail on the breadboard.

Circuit Diagram :-



Watchdog Demo starting
Watchdog Demo starting

Date : Expt. No. : Page No. : 20

Program :-

```
#include <avr/wdt.h>

void setup() {
  Serial.begin(9600);
  Serial.println("Watchdog Demo starting");
  pinMode(13, OUTPUT);
  wdt_disable();
  delay(3000);
  wdt_enable(WDTO_2S);
}

void loop() {
  for(int i=0; i<8; i++)
  {
    digitalWrite(13, HIGH);
    delay(10);
    digitalWrite(13, LOW);
    delay(10);
    wdt_reset();
  }
  while(1);
}
```

27/9/24

Result :- The configuration of watchdog timer in watchdog and interval mode is implemented successfully.

Date : 28/08/2024 Expt. No. : 5 Page No. : 11

Aim :-

To configure timer block for signal generation (with given frequency).

Components :-

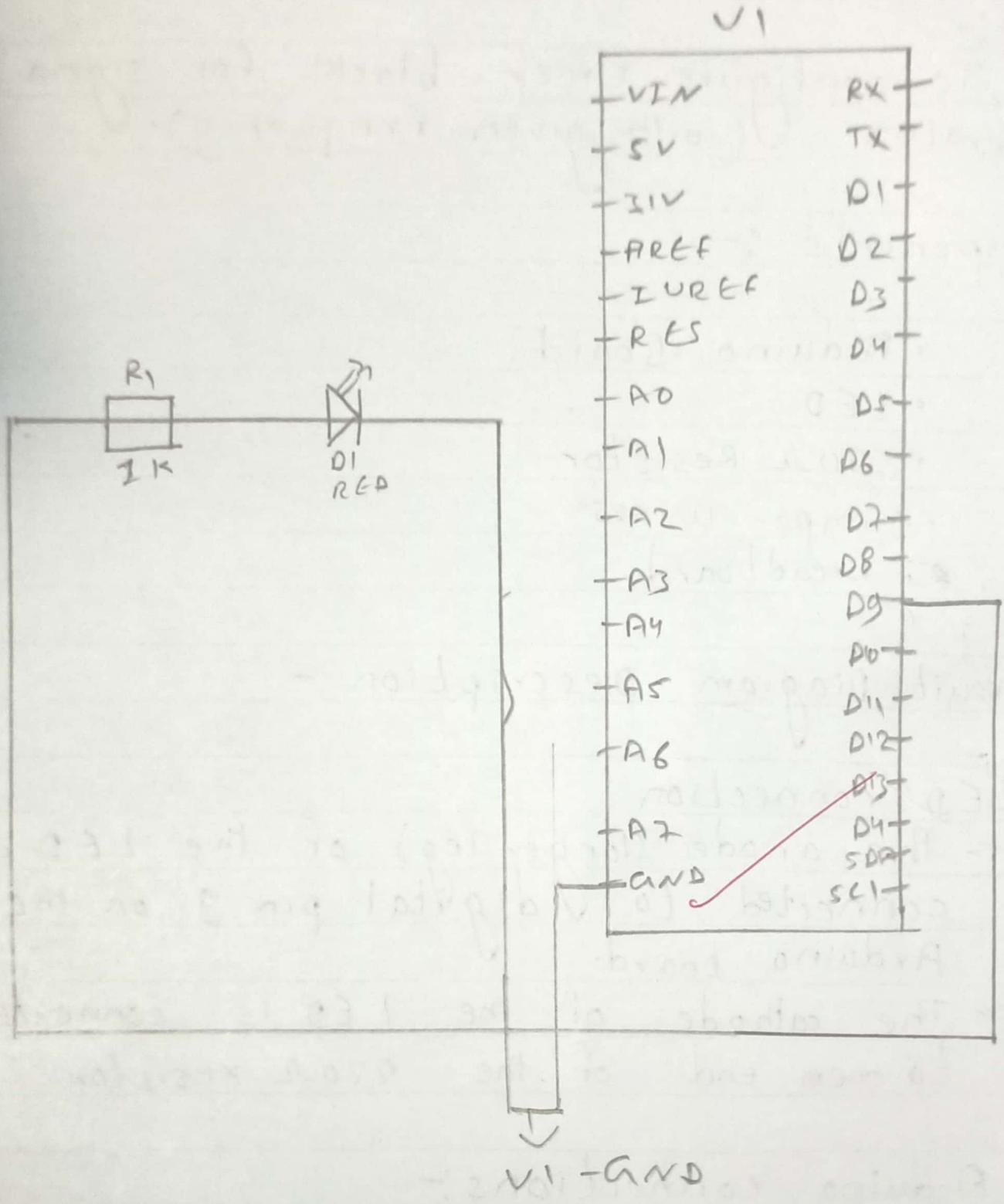
- Arduino Board
- LED
- 220Ω Resistor
- Jumper wires
- Breadboard

Circuit Diagram Description :-

- LED connection :
  - The anode (longer leg) of the LED is connected to digital pin 9 on the Arduino board.
  - The cathode of the LED is connected to one end of the 220Ω resistor.

- Arduino connections :-
  - connect the 5V pin of the Arduino to the power rail on the breadboard.
  - connect the GND pin.

Circuit Diagram:-



Date : Expt. No. : Page No. : 12

Program :-

```
const int outputPin = 9;
const long frequency = 1000;

void setup() {
  pinMode(outputPin, OUTPUT);
}

long timerCompare = (16000000 / (2*1*f))-1;

TCCR1A = 0;
TCCR1B = 0;
TCNT1 = 0;

OCR1A = timerCompare;
TCCR1B |= (1 << WGM12);
TCCR1B |= (1 << CS10);
TIMSK1 |= (1 << OCIE1A); }

void loop() { }

ISR(Timer1-compa-vect) {
  digitalWrite(outputPin, !digitalRead(outputPin));
}
```

Aim Result:- The observation of timer block signal is done successfully.

Date : 02/09/2024 Expt. No. : 6 Page No. : 23

Aim :-

To read temperature of MSP430 with the help of ADC.

Components Required:

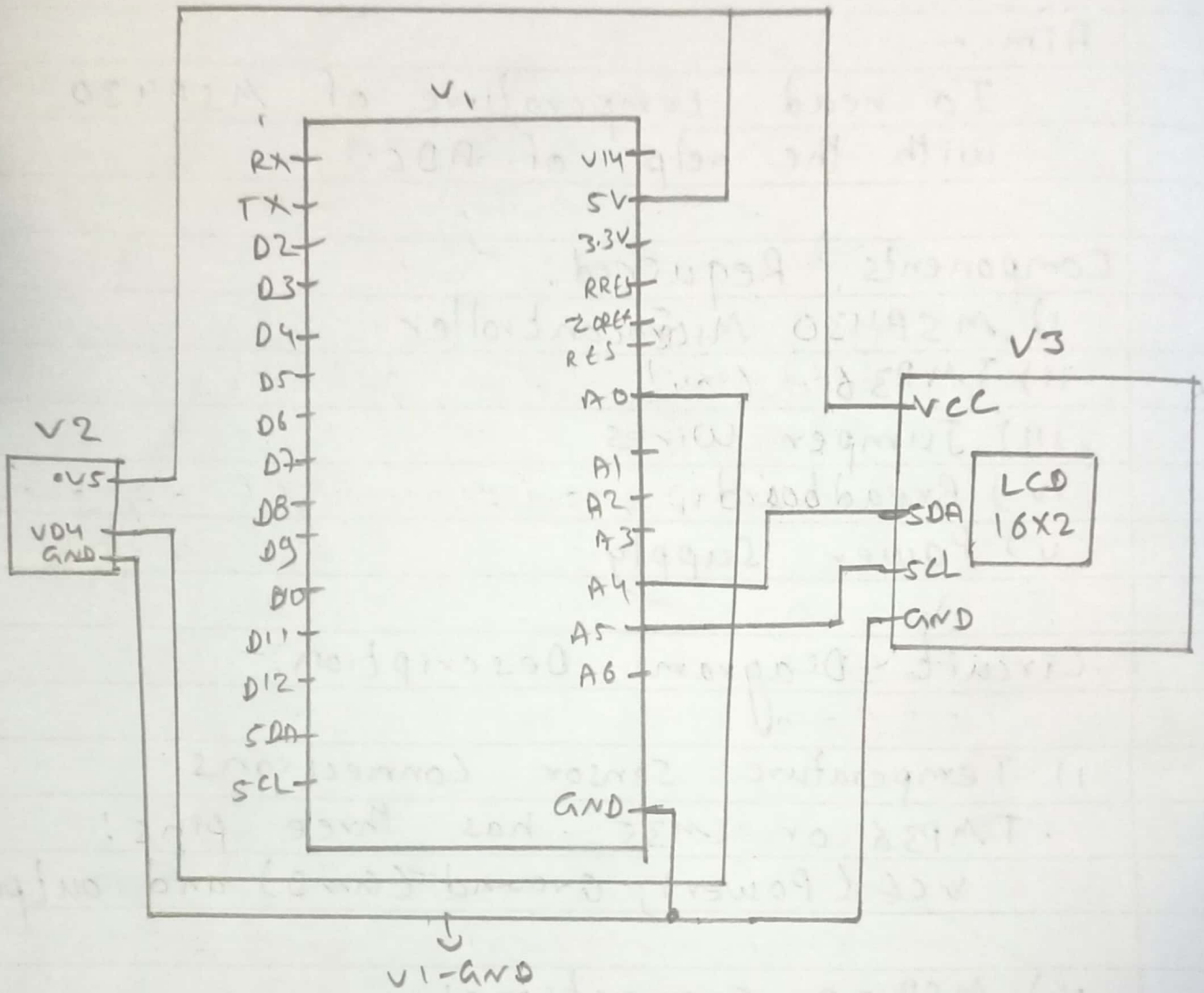
i) MSP430 Microcontroller
ii) TMP36
iii) Jumper Wires
iv) Breadboard
v) Power supply

Circuit Diagram Description:-

i) Temperature sensor Connections
- TMP36 or LM35 has three pins: VCC (Power), Ground (GND) and output.

ii) MSP430 connections:
- Connect the 3.3V pin
- Connect the GND pin of MSP430.

Circuit Diagram:-



. Program :-

```
const int sensorPin = A0;

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

void loop() {
int sensorValue = analogRead(sensorPin);
float voltage = sensorValue * (5.0/1023.0);
float temperatureC = (voltage - 0.5) * 100.0;

Serial.print("Temperature: ");
Serial.print(temperatureC);
Serial.println("°C");
delay(1000);
}
```

Result :-

- The reading of temperature of MSP430 with the help of ADC is done successfully.

Date : 04/09/2024 Expt. No. : 7 Page No. : 15

Aim:- To simulate and observe behaviour of Arduino when entering and existing power-down modes.

Components required:-

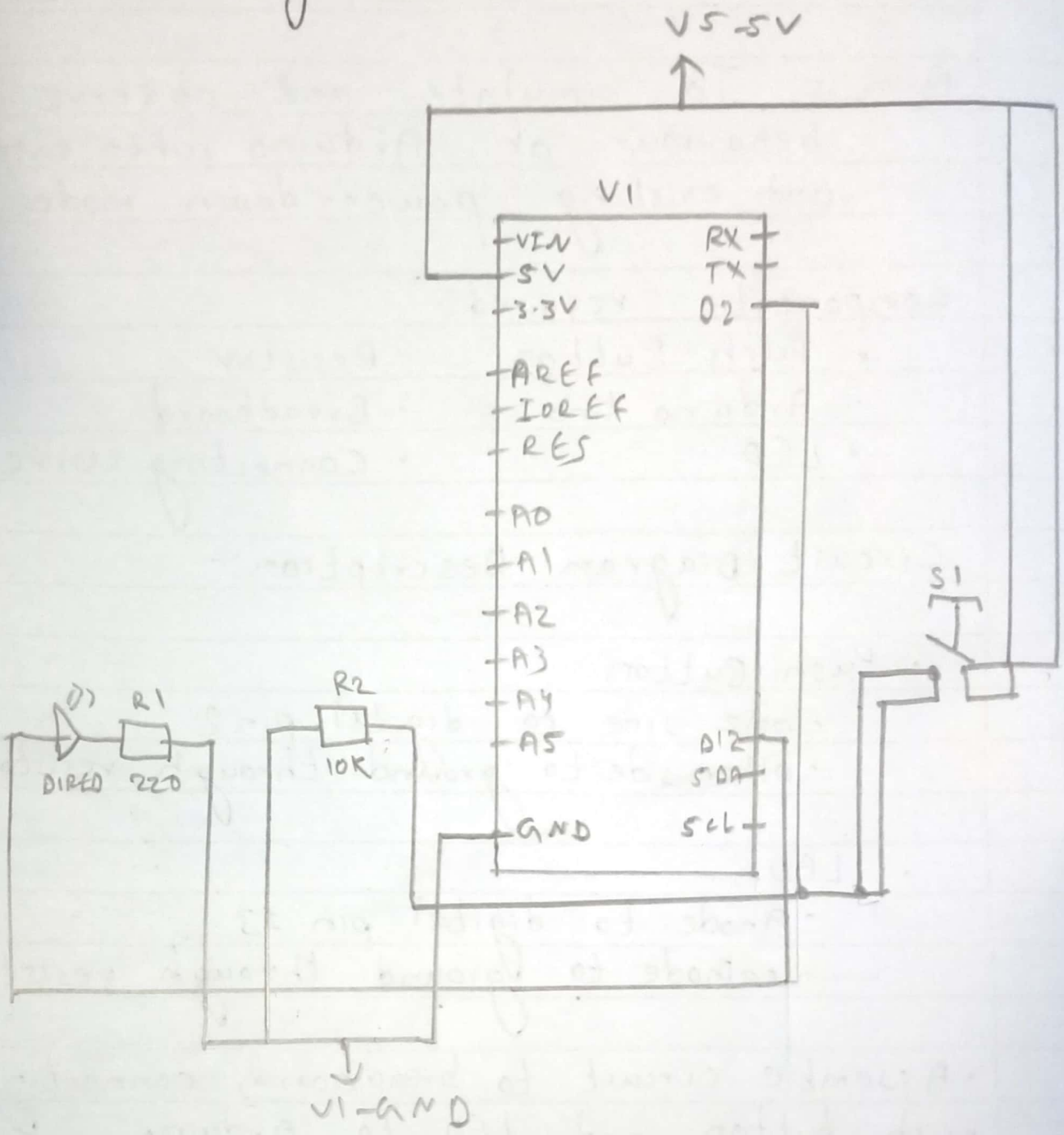
- Push Button
- Arduino Uno
- LED
- Resistor
- Breadboard
- Connecting Wire

Circuit Diagram Description:-

- Push Button:
  - one side to digital pin 2
  - other side to ground through resistor.
- LED
  - Anode to digital pin 13
  - cathode to ground through resistor.

- Assemble circuit to breadboard, connecting push button and LED to Arduino.
- Upload code to Arduino and Run the simulation.
- Press push button to trigger Arduino's power down mode.

Circuit diagram:-



Output: Entering Power-Down mode.

~~Entering~~ Exiting Power-Down mode.

Date : Expt. No. : Page No. : 16

Program :-

```
#include <avr/sleep.h>

int buttonPin = 2;
int ledPin = 13;

void setup(){
pinMode (buttonPin, INPUT);
pinMode (ledPin, OUTPUT);
Serial.begin(9600);
}
void loop () {
    if (digitalRead(buttonPin) == HIGH) {
    Serial.println("Entering Power-Down mode");
    digitalWrite(ledPin, HIGH);
    delay (500);
    set_sleep_mode (SLEEP_MODE_PWR_DOWN);
    sleep_enable();
    sleep_mode();
    sleep_disable();
    Serial.println ("Exiting Power-Down Mode");
    digitalWrite (ledPin, LOW);
    } delay (1000); }
```

Result :- The test of various power Down modes in Arduino is tested successfully.

Date : 09/09/2024 Expt. No. : 8 Page No. : 17

Aim :-

To generate PWM using Arduino and LED.

components required :-

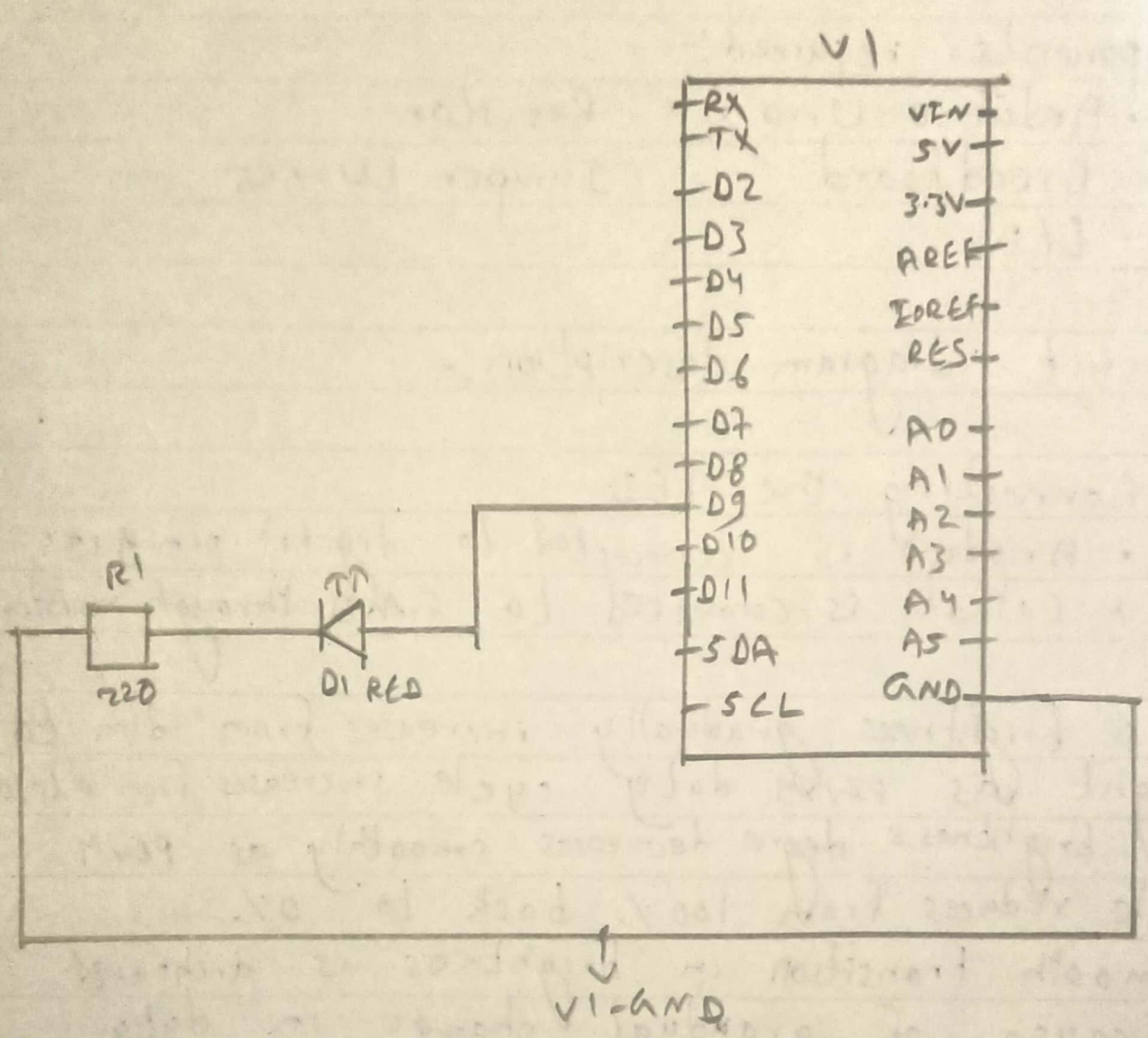
- Arduino Uno
- Resistor
- Breadboard
- Jumper Wires
- LED

circuit diagram description :-

- Connecting the LED
  - Anode is connected to digital pin 13.
  - Cathode is connected to GND through resistor.

- LED brightness gradually increases from dim to bright as PWM duty cycle increases from 0 to 100%.
- LED brightness ~~degre~~ decreases smoothly as PWM cycle reduces from 100% back to 0%.
- Smooth transition in brightness is archieved because of gradual change in duty cycle from 0 to 255.

Circuit Diagram :-



Output :-

Frequency set

Program :-

```
#include <smart-duty_cycling.h>

smart_duty_cycling cycle;
void setup(){
Serial.begin(9600);
cycle.setFrequencyDutyCycle(1,0.5);
  }
void loop (){
 if (cycle.switchMode())
 {
   if (cycle.wake)
   {
     digitalWrite(13, HIGH);
   }
   else {
     digitalWrite(13, LOW);
   } } }
```

Result :-

The experiment has successfully demonstrated use of PWM to control brightness of an LED.

Date : 23/09/2024 Expt. No. : 9 Page No. : 19

Aim :-

To use comparator to compare the signal threshold.

Components required:

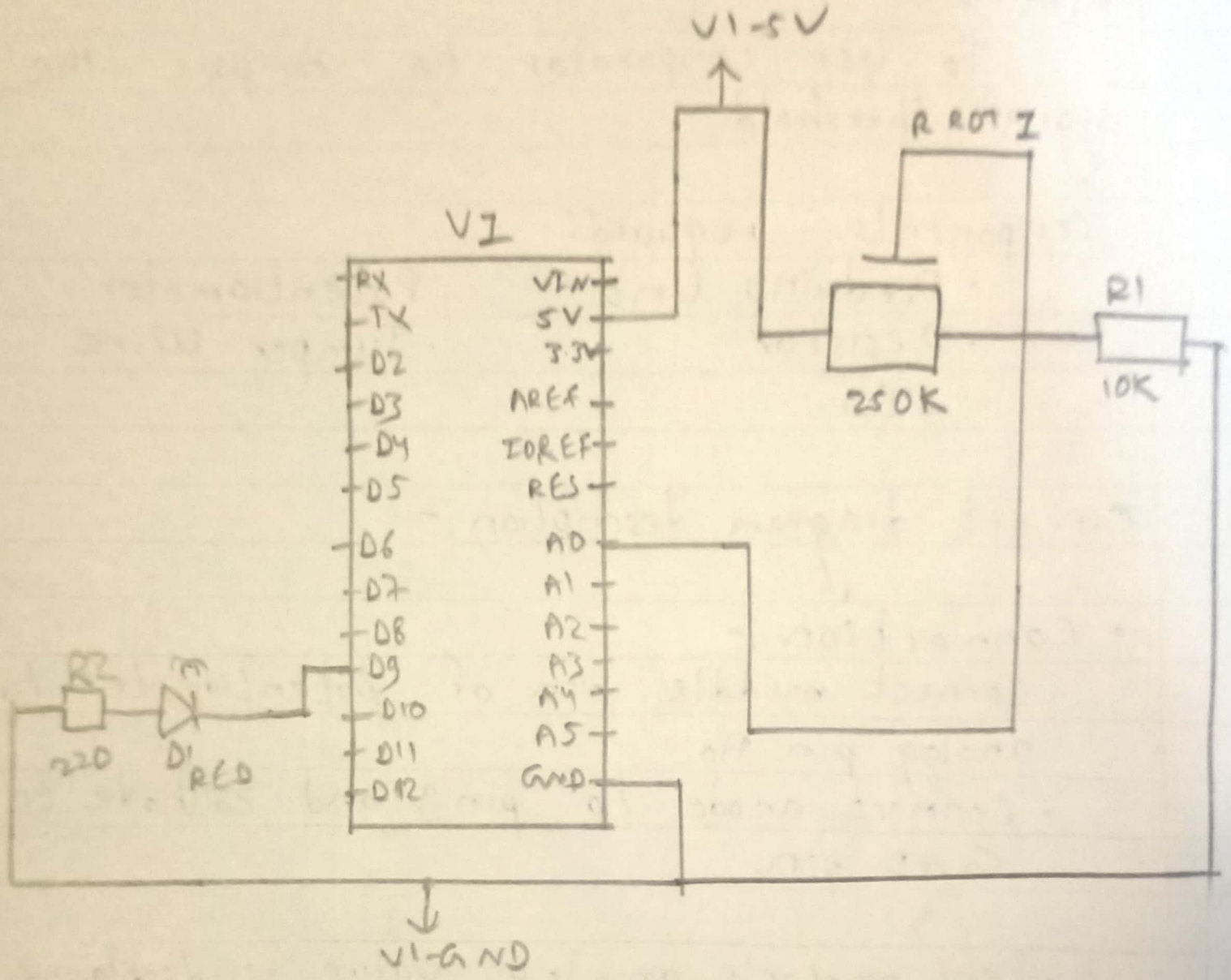
- Arduino Uno
- Potentiometer
- Resistor
- Jumper Wire
- LED

Circuit diagram description :-

- Connection :-
  - Connect middle pin of potentiometer to analog pin A0.
  - Connect anode to ping and cathode to GND pin.

- Potentiometer's analog value is displayed.
- LED turns on when value is greater than 512
- LED turns off when value is below 512.

Circuit Diagram:-



Output:- Sensor value: 510
Sensor value: 527

Date : Expt. No. : Page No. : 20

Program :-

```
int sensorPin = A0;
int ledPin = 9;
int threshold = 512;
void setup(){
 pinMode(ledPin, OUTPUT);
 Serial.begin(9600);
 }
void loop() {
 int sensorValue = analogRead(sensorPin);
 if (sensorValue > threshold) {
  digitalWrite(ledPin, HIGH);
 } else {
 digitalWrite(ledPin, LOW);
 }
 serial.println("Sensor value:");
 Serial.println(sensorValue);
 delay(100);
 }
```

Result :-

The use of comparator to compare the signal threshold. is done successfully.

Date : 25/09/2024 Expt. No. : 10 Page No. : 21

Aim :-

To control the speed of DC Motor.

Components required :-

- Arduino Uno
- 9V Battery
- DC motor.
- L293D Motor
- Jumper Wires

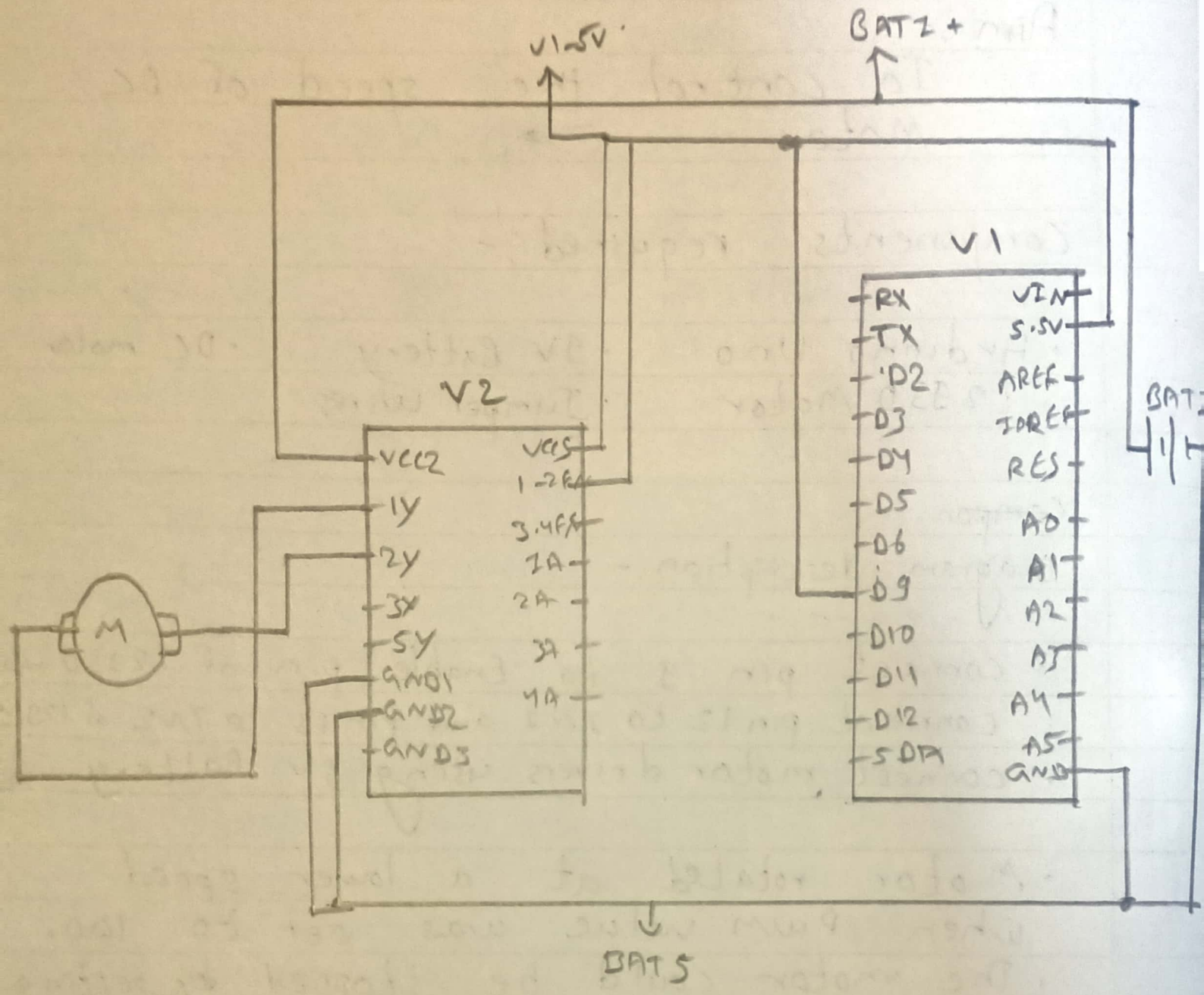
~~Compon~~

Diagram description :-

- connect pin 9 to Enable pin of L293D motor.
- connect pin 12 to IN1 and Pin 13 to IN2 of L293D.
- connect motor drivers using 9V battery.

- Motor rotated at a lower speed when Pwm value was set to 100.
- The motor could be stopped by setting Pwm value to 0.
- By changing direction pins, motor direction could be reversed.

Circuit Diagram:-



Output:- Clockwise

~~An~~ Counter clock wise

Date : Expt. No. : Page No. : 22

Program :-

```
#include <stepper.h>
const int stepsPerRevolution = 200;
stepper mystepper (stepsPerRevolution, 8,9,10);
void setup(){
  mystepper.setSpeed (60);
  Serial.begin (9600);
}
void loop() {
  Serial.println ("clockwise");
  mystepper.step (stepsPerRevolution);
  delay (500);

  Serial.println ("counterclockwise");
  mystepper.step (-stepsPerRevolution);
  delay (500);
}
```

Result :-

The speed of DC motors is controlled successfully.

Date : 07/10/2024 Expt. No. : II Page No. : 23

Aim :-

To implement Master slave communication between Msp's using Wireh

Components used :-

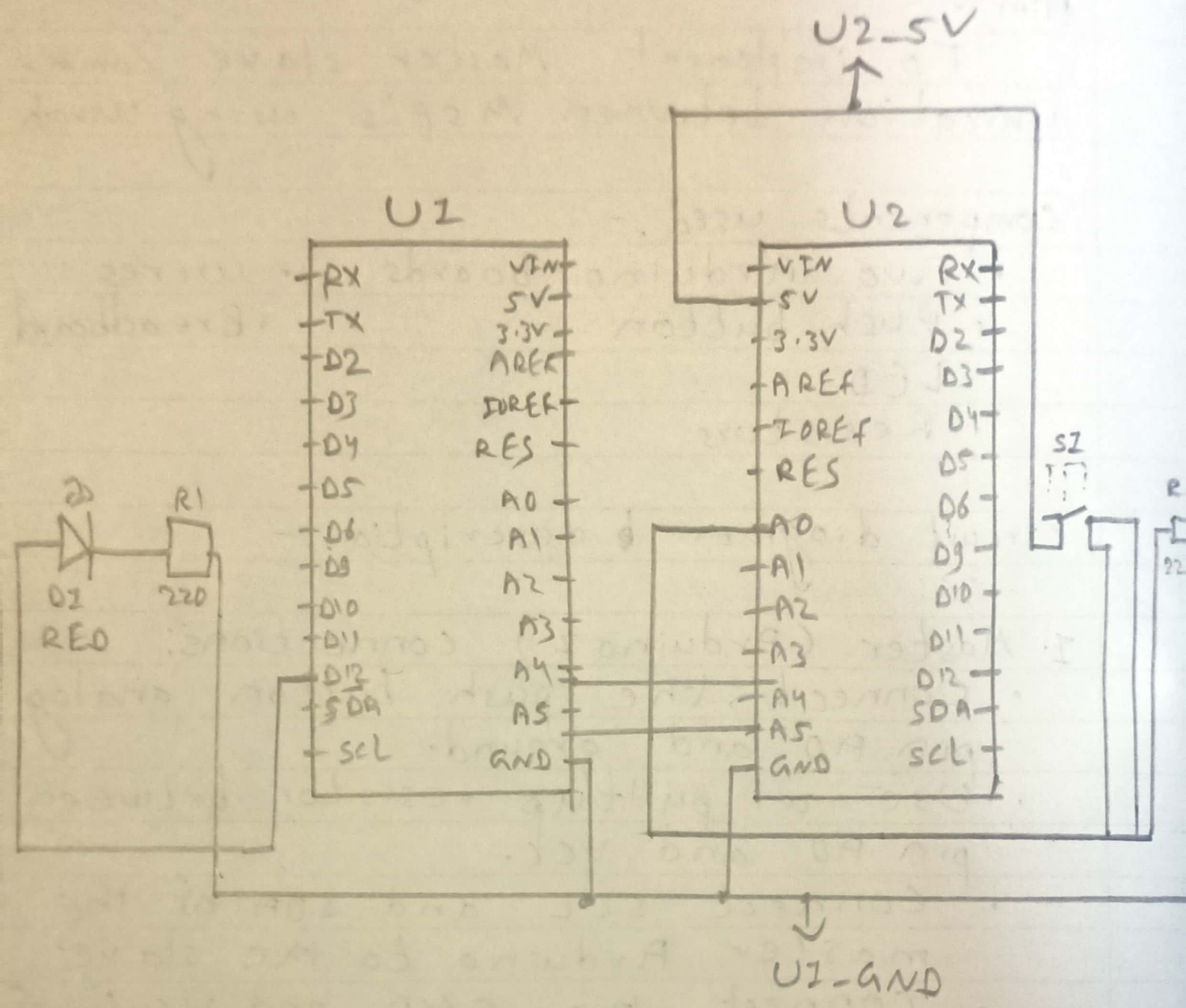
- Two Arduino boards
- wires
- Push button
- Breadboard
- LED
- Resistors

Circuit diagram & description :-

1. Master (Arduino 1) connections:
   - Connect the push button analog pin A0 and ground.
   - Use a pull-up resistor between pin A0 and VCC.
   - Connect SCL and SDA of the master Arduino to the slave.
   - Connect the GND and VCC of both Arduinos together for proper power and signal reference.

2. Slave (Arduino2) connections :-
   - Connect the LED to digital pin 13 with a 220Ω or 330Ω resistor in series with the LED and ground.

Circuit Diagram:-



OUTPUT : RED Light glow

Code For Arduino 1 :-

```
#include <Wire.h>
int pushButton = A0;
int x = 0;
void setup()
{
 wire.begin();
 pinMode(pushButton, INPUT);
 }
void loop()
{
  wire.beginTransmission(1);
  x = digitalRead(pushButton);
  Wire.write(x);
  Wire.endTransmission();
  delay(500);
 }
```

Code for Arduino 2;

```
#include <Wire.h>
int pinLed = 13;
int x=0;
void setup()
 {
  Wire.begin(1);
```

```
Wire.onReceive(receiveEvent);
pinMode(pinLed, OUTPUT);
}
void loop()
{
  delay(100);
}
void receiveEvent(int howMany){
x = Wire.read();
if (x == 1){
 digitalWrite(pinLed, HIGH);
}
else{
  digitalWrite(pinLed, LOW);
}}
```

Result :-

Master slave communication between MSPs using wire.h is implemented successfully.

Date : 09/10/2024 Expt. No. : 12 Page No. : 26

Aim :-

To implement Networking MSPs using two Arduinos.

Component required :-

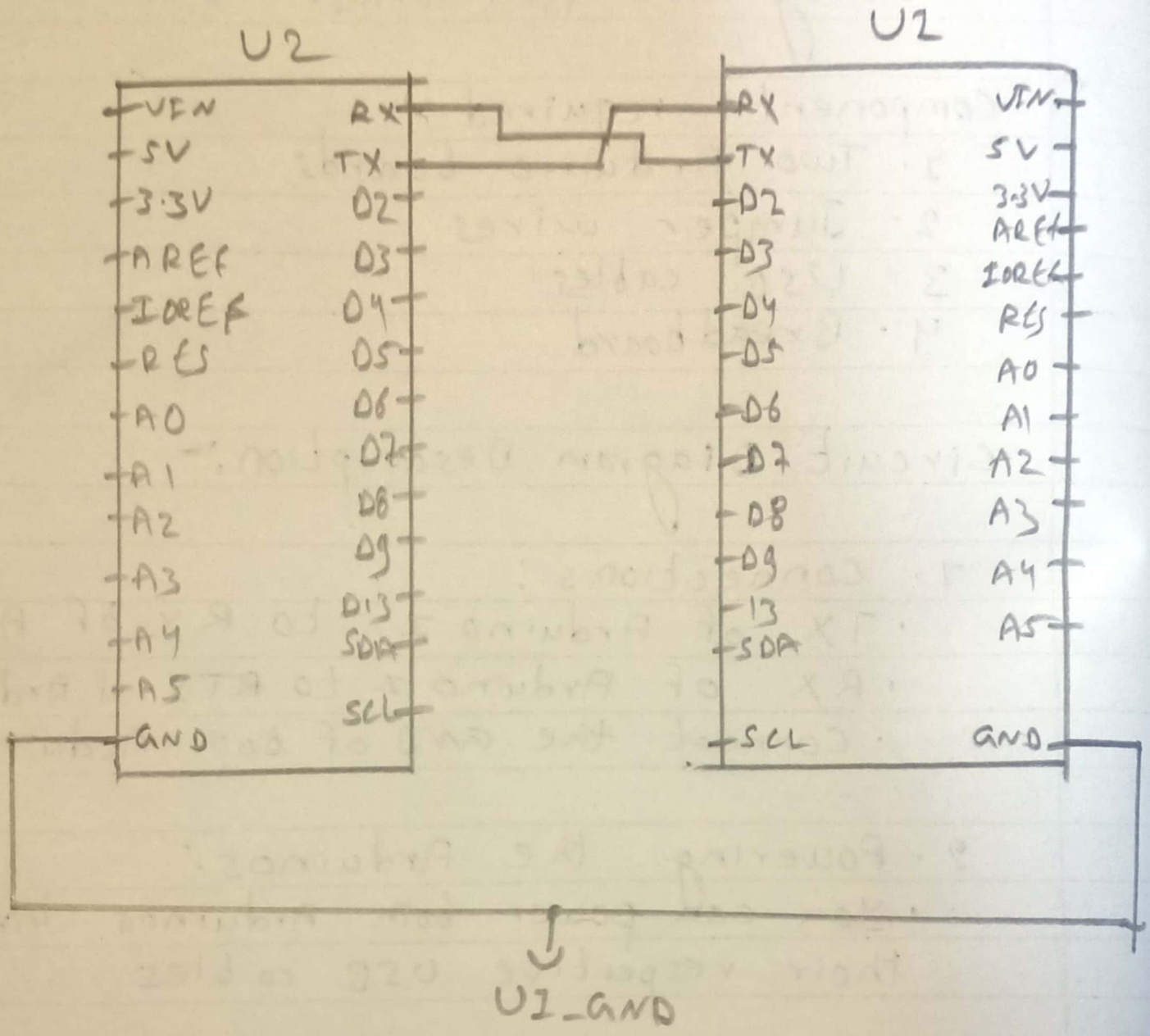
1. Two Arduino boards
2. Jumper wires
3. USB cables
4. Breadboard

Circuit Diagram Description :-

1. Connections :
   - TX of Arduino 1 to RX of Arduino 2.
   - RX of Arduino 1 to TX of Arduino 2
   - Connect the GND of both Arduinos.

2. Powering the Arduinos :
   - You can power both Arduinos through their respective USB cables.

Circuit Diagram:-



OUTPUT:- Hello Arduino 2!

-Received : Hello Arduino 2!

Code for Arduino 1 :-

```
void setup() {
  Serial.begin (9600);
}
void loop (){
  Serial.println ("Hello Arduino 2");
  delay (2000);
}
```

Code for Arduino 2 :-

```
String receiveMessage = "";
void setup(){
  Serial.begin(9600);
  Serial.println ("Waiting for data");
}
void loop (){
if (Serial.available() > 0){
char incomingByte = Serial.read();
if (incomingByte == "\n" || incomingByte == '\r')
{
  if (receivedMessage.length () > 0){
  Serial.print ("Received:");
  Serial.println (receivedMessage);
  receivedMessage = "";
```

```
        }
      }
      else {
      receivedMessage += incomingByte;
    }
  }
}
```

Result :-

The implementation of Networking MSPs using two Arduinos. is done successfully.