

**PROJECT REPORT**

**Computer Organization and Assembly Language (COAL)**

**Light Detection System**

**Submitted To:**

Sir Majid Shafique

**Submitted By:**

* Aima malik (001)
* Ayesha Nadeem (020)
* Khadija Shafqat (051)
* Mahnoor sajjad (054)

**Light Detection System Using LDR**

## **LISTS OF COMPONENTS:**

* Arduino UNO
* MAX 232
* LDR
* Jumper wires
* Breadboard
* Resistor 10k ohm
* Resistor 220 ohm
* Led
* Rs-232 cable
* **Code of .ino:**

extern "C" {

void adc\_init();

uint16\_t adc\_read();

}

const int ledPin = 13;

const int threshold = 400;

void setup() {

Serial.begin(9600);

adc\_init();

pinMode(ledPin, OUTPUT);

}

void loop() {

uint16\_t sensorValue = adc\_read();

if (sensorValue < threshold) {

digitalWrite(ledPin, HIGH);

Serial.println("Dark detected");

} else {

digitalWrite(ledPin, LOW);

Serial.println("Light detected");

}

delay(500);

}

* **Code of .s:**

/\* Assembly code for AVR ATmega328p ADC init and read \*/

.equ ADMUX , 0x7C ; ya voltage decide karta hai

.equ ADCSRA , 0x7A ;ya adc control karta hai

.equ ADCL , 0x78 ; low

.equ ADCH , 0x79 ; high

.equ REFS0 , 6

.equ ADEN , 7

.equ ADSC , 6

.equ ADPS2 , 2

.equ ADPS1 , 1

.equ ADPS0 , 0

.section .text

.global adc\_init ; function start karta hai adc ko initialized karta hai

.global adc\_read

; void adc\_init()

adc\_init:

; Set ADMUX to AVcc with external capacitor on AREF pin (1 << 6) = 0x40

ldi r16, 0x40

sts ADMUX, r16

; Enable ADC (1 << 7) + prescaler 128 (ADPS2=1<<2, ADPS1=1<<1, ADPS0=1<<0)

; 0x80 + 0x04 + 0x02 + 0x01 = 0x87

ldi r16, 0x87

sts ADCSRA, r16

ret

; uint16\_t adc\_read()

adc\_read:

; Start conversion by setting ADSC bit (1 << 6 = 0x40)

lds r16, ADCSRA

ori r16, 0x40

sts ADCSRA, r16

wait\_conversion:

lds r16, ADCSRA

sbrc r16, ADSC ; skip next if ADSC bit cleared

rjmp wait\_conversion

; Read ADCL first

lds r24, ADCL

; Read ADCH next

lds r25, ADCH

ret

* **Working:**

**Hardware Setup:**

The hardware setup of the Light Detection System includes the following components connected on a breadboard and controlled by an Arduino UNO:

* **Arduino UNO**: Acts as the main controller. Reads data from the LDR and controls the LED and serial output.
* **LDR (Light Dependent Resistor)**: Detects the intensity of light in the environment.
* **10kΩ Resistor**: Used with the LDR to form a voltage divider circuit to produce a variable voltage based on light.
* **LED**: Indicates the presence or absence of light. Connected through a 220Ω resistor to limit current.
* **220Ω Resistor**: Protects the LED from excess current.
* **MAX232 IC**: Converts TTL logic signals from Arduino to RS-232 voltage levels.
* **RS-232 Cable**: Transfers serial data from the MAX232 to a PC or other serial device.
* **Breadboard and Jumper Wires**: Used to build the circuit without soldering.

**Working Process:**

1. **Light Sensing:**
   * The **LDR** changes its resistance depending on the amount of light falling on it.
   * It is connected in a **voltage divider** circuit with the **10kΩ resistor**.
   * The output voltage from the divider is fed into an **analog pin (e.g., A0)** of the **Arduino UNO**.
2. **Analog to Digital Conversion:**
   * The Arduino reads the analog voltage using analogRead().
   * This voltage corresponds to light intensity:
     + **Low voltage** → **Bright light**
     + **High voltage** → **Darkness**
3. **Decision Making:**
   * The Arduino compares the LDR reading against a **predefined threshold**.
   * If the reading indicates **darkness**, it:
     + Turns **ON the LED**.
     + Sends a message **"Light Detected: OFF"** via serial communication.
   * If the reading indicates **light**, it:
     + Turns **OFF the LED**.
     + Sends a message **"Light Detected: ON"** via serial.
4. **Serial Communication:**
   * Arduino sends messages through its **TX pin**.
   * These signals are converted by the **MAX232 IC** to RS-232 voltage levels.
   * The **RS-232 cable** connects the output of the MAX232 to a PC (COM port).
   * Messages can be read using a **Serial Monitor application (e.g., PuTTY, Arduino Serial Monitor)**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_