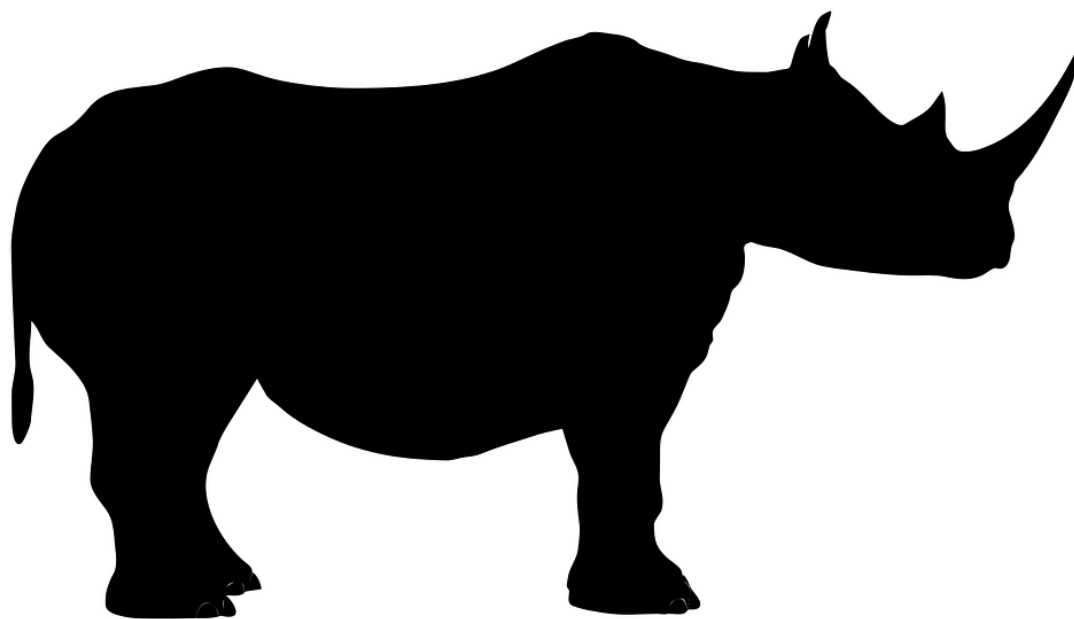


# PROJECT RHINO



**Description :** 8051 Hardware Abstraction Layer

**Synopsis :** Project Rhino is a package of set of header files to abstract the 8051 hardware from the user

**Contents :**

- Communication header file
- Timer header file
- Interrupt header file
- Power control

# Contents

<i>1. List of functions</i>	<i>- 03</i>
<i>2. Communication</i>	<i>- 06</i>
<i>3. Timers</i>	<i>- 07</i>
<i>4. Interrupts</i>	<i>- 10</i>
<i>5. Power Control</i>	<i>- 12</i>

# List of Functions

## 1. **Communication :**

- void serial\_trans(int baud,int Data)
- int serial\_rec(int baud)

## 2. **Timers:**

- void delay\_ms(int ms)
- void delay\_us(int us)
- void delay(int sec)
- int counter\_0()
- int counter\_1()

## 3. **Interrupts :**

- void ext\_0()
- void ext\_1()
- void ext\_priority(int a)

#### **4. Power control :**

- `void power_idle()`
- `void power_off()`



# Communication

## 1. Transmitter :

- Syntax : `serial_trans(int baud,int Data)`
- Description : Performs serial transmission of data . Takes 2 inputs as arguments with first as baud rate and second as the data.
- Example : `serial_trans(-3,'3');`

## 2.Receiver :

- Syntax : `serial_rec(int baud)`
- Description : Performs serial receival of data . Takes baud rate as input . Gives data as output
- Example : `serial_rec(-3);`

# Timers

## 1. Millisecond timer :

- Syntax : `delay_ms(int ms)`
- Description : Used to produce delay in milliseconds . Takes time in milliseconds as input .
- Example : `delay_ms(10);`

## 2. Microsecond timer :

- Syntax : `delay_us(int us)`
- Description : Used to produce delay in microseconds . Takes time in microseconds as input .
- Example : `delay_us(10);`



### 3. Seconds timer :

- Syntax : `delay(int sec)`
- Description : Used to produce delay in seconds .  
Takes time in seconds as input .
- Example : `delay(10);`

### 4. Counter 0 :

- Syntax : `counter_0()`
- Description : Used to count the external stimuli at  
pin 4 of port 3.
- Example : `counter_0();`

## 5. Counter 1 :

- Syntax : `counter_1()`
- Description : Used to count the external stimuli at pin 5 of port 3.
- Example : `counter_1();`

# Interrupts

## 1. External Interrupt 0 :

- Syntax : `ext_0()`
- Description : Used to respond to external interrupt at pin 2 at port 3.
- Example : `ext_0();`

## 2. External Interrupt 1 :

- Syntax : `ext_1()`
- Description : Used to respond to external interrupt at pin 3 at port 3.
- Example : `ext_1();`

### 3. Interrupt Priority :

- Syntax : `ext_priority(int a);`
- Description : Used to prioritize between external interrupt 0 and external interrupt 1. It takes external interrupt number as priority.
- Example : `ext_priority(1);`

# Power Control

## 1. Idle Mode :

- Syntax : `power_idle()`
- Description : Used to send microcontroller to idle mode.
- Example : `power_idle();`

## 2. Power Off Mode :

- Syntax : `power_off()`
- Description : Used to switch off the microcontroller
- Example : `power_off();`