EXPERIMENT NO:

ROLL No:

NAME:

TITLE: Implement power management in any embedded core of your choice.

Theory:

The modern electronic devices are constantly in search of more efficiency in all aspects. One of the most important criteria that enhances the value of a device is the amount of power consumed by it. With device manufacturers moving towards an environmentally friendly green computing and clean energy concepts, it becomes imperative to design circuits that are power efficient. For example, a standalone battery-powered device such as a smartphone, one of the key factor that a customer will explore before buying will be battery life. Hence power management while designing an embedded system is essential to utilize the available power efficiently.

Need for Power Optimization:

Size and Life of battery:

With miniaturization of devices and the emergence of Nano-technology, the space that a device can allot for a battery is reducing. The device is expected to efficiently utilize the available power and function without any trade-off in performance. Life of the battery comes in as an advantage during the commercialization of a device.

Performance and Power Utilization:

Embedded Systems are the crux of current technology and hence more functionalities added to the existing systems every day. Hence it becomes crucial to manage power. User usage of a device is of an unpredictable nature and hence the devices are designed to meet all types of workloads.

Adaptation of Green Technologies:

Power for future of electronic devices are likely to be obtained from greener technologies such as solar power. Solar power though, ecologically sustainable lacks the same power density as normal power. In such cases, devices are expected to work efficiently given any situation. Further power lost as heat has to be minimized.

Power consumption is a critical issue for a device running continuously for a long time without being turned off. So to overcome this problem almost every controller comes with a sleep mode, which help developers to design electronic gadgets for optimal power consumption. Sleep mode puts the device in power saving mode by turning off the unused module.

Code:

```
#include <dht.h>
#define dataPin 2
```

```
dht DHT;
void setup() {
Serial.begin(9600);
pinMode(13,OUTPUT);
digitalWrite(13,LOW);
}

void loop() {
digitalWrite(13,HIGH);
int readData = DHT.read11(dataPin); // DHT11
float t = DHT.temperature;
float h = DHT.humidity;
Serial.print("Temperature = ");
Serial.print(t);
Serial.print(" C | ");
Serial.print("Humidity = ");
Serial.print(h);
Serial.println(" % ");
delay(2000);
}
```

Arduino weather station with sleep mode:

```
#include <dht.h>
#include <LowPower.h>
#define dataPin 2
dht DHT;
void setup() {
Serial.begin(9600);
pinMode(13,OUTPUT);
digitalWrite(13,LOW);
}

void loop() {
Serial.println("Get Data From DHT11");
delay(1000);
digitalWrite(13,HIGH);
int readData = DHT.read11(dataPin); // DHT11
float t = DHT.temperature;
float h = DHT.humidity;
Serial.print("Temperature = ");
Serial.print(t);
Serial.print(" C | ");
```

```
Serial.print("Humidity = ");
Serial.print(h);
Serial.println(" % ");
delay(2000);
Serial.println("Arduino:- I am going for a Nap");
delay(200);
digitalWrite(13,LOW);
LowPower.idle(SLEEP_8S, ADC_OFF, TIMER2_OFF, TIMER1_OFF, TIMER0_OFF, SPI_OFF,
USART0_OFF, TWI_OFF);
Serial.println("Arduino:- Hey I just Woke up");
Serial.println("");
delay(2000);
}
```

Result: