```ino
#include <LowPower.h>
#include <RCSwitch.h>

#define NUM_SAMPLES 500
#define LDR_THRESHOLD 600
#define LDR_VIN_PIN 7
#define LDR_DATA_PIN 5
#define TX_VIN_PIN 8
#define TX_DATA_PIN 10
#define TIME_SLEEPING 30
#define NODE_ID 0

// Instancia objeto de comunicação
RCSwitch mySwitch = RCSwitch();

float get_ldr_value() {
  // Obtém amostras do LDR
  float status = 0.0;
  for (int i = 0; i < NUM_SAMPLES; i++)
    status += analogRead(LDR_DATA_PIN);

  return status / NUM_SAMPLES;
}

void read_and_send_data() {
  // Obtém medidas do sensor
  digitalWrite(LDR_VIN_PIN, HIGH);
  float ldr_measure = get_ldr_value();

  // Transmite os dados
  digitalWrite(TX_VIN_PIN, HIGH);
  mySwitch.send(NODE_ID, 24);
  mySwitch.send(int(ldr_measure), 24);
}

void setup() {
  // Configura pinos de alimentação do sensor e RF
  pinMode(LDR_VIN_PIN, OUTPUT);
  pinMode(TX_VIN_PIN, OUTPUT);

  // Configura pino de dados do RF
  mySwitch.enableTransmit(TX_DATA_PIN);
}

void loop() {
  // Permanece em modo de baixo consumo durante 'TIME_SLEEPING' segundos
  for (int i = 0; i < TIME_SLEEPING; i++)
    LowPower.powerDown(SLEEP_1S, ADC_OFF, BOD_OFF);

  // Sensor realiza leitura e transmite estado da vaga
  read_and_send_data();

  // Volta ao modo de baixo consumo
  digitalWrite(LDR_VIN_PIN, LOW);
  digitalWrite(TX_VIN_PIN, LOW);
}
```