

## Binary Semaphore

```
#include <Arduino_FreeRTOS.h>
#include <semphr.h>
SemaphoreHandle_t xBin_Semaphore;
TaskHandle_t myTask1=NULL;
TaskHandle_t myTask2=NULL;
TickType_t t;

void setup()
{
    Serial.begin(9600); // Enable serial communication library.
    while(!Serial)
    {
        ;
    }

    // Create task for Arduino led
    xTaskCreate(Task1, // Task function
        "Ledon", // Task name
        128, // Stack size
        NULL,
        3, // Priority
        &myTask1 );
    xTaskCreate(Task2, // Task function
        "Ledoff", // Task name
        128, // Stack size
        NULL,
        2, // Priority
        &myTask2);
    xBin_Semaphore = xSemaphoreCreateBinary();
    // xSemaphoreGive(xBin_Semaphore);

}

void loop() {}

void Task1(void *pvParameters)
{
    (void) pvParameters;

    for (;;)
    {
        Serial.println("Task1 start");
        digitalWrite(LED_BUILTIN, LOW);
        xSemaphoreGive(xBin_Semaphore);
    }
}
```

```

    t = xTaskGetTickCount();
    Serial.print("\tAt time = ");
    Serial.println(t);
    Serial.println("Task1 gave semaphore");
    vTaskDelay(200);
}
}

void Task2(void *pvParameters)
{
    (void) pvParameters;
    for (;;)
    {
        if (xSemaphoreTake(xBin_Semaphore, portMAX_DELAY))
        {
            Serial.println("Inside Task2 ");
            t = xTaskGetTickCount();
            Serial.print("\tAt time = ");
            Serial.println(t);
            digitalWrite(LED_BUILTIN, HIGH);
        }
        else
        {
            Serial.println("Inside Task2 No Success");
            t = xTaskGetTickCount();
            Serial.print("\tAt time = ");
            Serial.println(t);
        }
    }
    vTaskDelay(100);
}
}

```