|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| C:\Users\SebSan\Pictures\unach.jpg  **UNIVERSIDAD NACIONAL DE CHIMBORAZO**  **FACULTAD DE INGENIERIA** | | | | | | |  |
| **GUÍA DE PRÁCTICAS**  **PERIODO ACADÉMICO: MAYO 2020 - OCTUBRE 2020** | | | | | | | **VERSIÓN:** 1 |
| **Página 1 de 8** |
| **CARRERA:**  **TELECOMUNICACIONES** | | **DOCENTE:**  **LEONARDO RENTERIA** | | **SEMESTRE: CUARTO**  **PARALELO: A** | | | |
| **NOMBRE DE LA ASIGNATURA:**  DISPOSITIVOS EMBEBIDOS Y LAB. | | **CÓDIGO DE LA ASIGNATURA:**  EETOP10B | | **LABORATORIO A UTILIZAR:**  ELECTRONICA | | | |
| **Práctica No.:**  **3** | **Tema:**  Aplicación de los Sistemas Embebidos | | Duración (horas)  2 | | No. Grupos  6 | No. Estudiantes (por Grupo)  Paul Maldonado  Mauricio Pilatuña  Cinthya Pilco  Jeffersson Pino | |
| **Objetivos de la Práctica:**  Implementar un sistema RTOS en un microcontrolador | | | | | | | |
| **Equipos, Materiales e Insumos:**   * **Simulador** * **Osciloscopio, Generador de funciones,** * **Multímetro,** * **Arduino,** * **Pc** * **LEDs** * **Potenciómetro** * **Resistencias** | | | | | | | |
| **Procedimiento:**   1. Arme el circuito de la figura       **Fig 1: Circuito implementado.**    **Fig 2: Circuito Simulado (ISIS Proteus).**   1. Implementar un sistema RTOS tipo **LOOP** que permita encender los LEDs cada 100 y 500 ms.   #define Ledazul 3  #define Ledverde 5  #define Potenciometro A0  void setup() {  Serial.begin(9600);  pinMode(Ledazul, OUTPUT);  pinMode(Ledverde, OUTPUT);  }  void loop()  {  Parpadeo100();  Parpadeo500();  Leer();  }  void Parpadeo100(){  digitalWrite(Ledazul, HIGH);  digitalWrite(Ledazul, LOW);  delay(100);  }  void Parpadeo500(){  digitalWrite(Ledverde, HIGH);  digitalWrite(Ledverde, LOW);  delay(500);  }  void Leer(){  int Valor = analogRead(Potenciometro);  Serial.println(Valor);  delay(1000);  }   1. Probar el funcionamiento de usando el osciloscopio     **Fig 3: Resultado simulación sistema LOOP (Osciloscopio).**   1. Implementar un sistema RTOS usando la librería FREERTOS para encender los LEDs cada 100 y 500 ms, lea el canal analógico A0 y envíe los datos por el puerto serial. Crear una tarea nueva para cada uno de ellos.   #include <Arduino\_FreeRTOS.h>  #define Ledazul 3  #define Ledverde 5  #define Potenciometro A0  void Parpadeo100( void \*pvParameters );  void Parpadeo500( void \*pvParameters );  void Leer( void \*pvParameters );  void setup() {  xTaskCreate(  Parpadeo100  , (const portCHAR \*) "parpadeo1"  , 128  , NULL  , 2  , NULL );  xTaskCreate(  Parpadeo500  , (const portCHAR \*) "parpadeo2"  , 128  , NULL  , 3  , NULL );  xTaskCreate(  Leer  , (const portCHAR \*) "lectura"  , 128  , NULL  , 1  , NULL );    }  void loop()  {    }  void Parpadeo100(void \*pvParameters)  {  (void) pvParameters;    pinMode(Ledazul, OUTPUT);  for (;;)  {  digitalWrite(Ledazul, HIGH);  vTaskDelay( 10 / portTICK\_PERIOD\_MS );  digitalWrite(Ledazul, LOW);  vTaskDelay( 90 / portTICK\_PERIOD\_MS );  }  }  void Parpadeo500(void \*pvParameters)  {  (void) pvParameters;    pinMode(Ledverde, OUTPUT);  for (;;)  {  digitalWrite(Ledverde, HIGH);  vTaskDelay( 10 / portTICK\_PERIOD\_MS );  digitalWrite(Ledverde, LOW);  vTaskDelay( 490 / portTICK\_PERIOD\_MS );  }  }  void Leer(void \*pvParameters)  {  (void) pvParameters;  Serial.begin(9600);  for (;;)  {    int Valor = analogRead(Potenciometro);  Serial.println(Valor);  vTaskDelay(1000/ portTICK\_PERIOD\_MS);  }  }   1. Probar el funcionamiento de usando el osciloscopio.     **Fig 4: Resultado simulación sistema RTOS (Osciloscopio).**    **Fig 5: Lectura de datos puerto A0 (virtual terminal).** | | | | | | | |
| **Resultados:**  Conocer el funcionamiento de los sistemas RTOS  El sistema RTOS o sistema operativo en tiempo real administra los recursos del hardware por ejemplo la gestión de memoria, registros, etc. El uso de un RTOS fue indispensable para el correcto funcionamiento de lo solicitado ya que las tareas se crean de manera independiente y el scheduler administra el tiempo de ejecución de cada uno según la prioridad de estas, los procesos se realizan de manera precisa a diferencia del uso de un bucle loop donde iniciaríamos las tareas de forma conjunta y el timer que uno le administrara no contara con la precisión necesaria.  Comparando los resultados de los 2 ejercicios podemos observar claramente que el sistema RTOS es mejor en la ejecución. | | | | | | | |
| **Anexos:**  https://github.com/JeffPino/Sistemas-Embebidos.git | | | | | | | |
| **Referencias bibliográficas:** | | | | | | | |

**Fecha de Revisión y Aprobación**: 18/05/2020

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

**Firma Director de Carrera Firma Docente**