

UNIVERSIDAD POLITÉCNICO GRAN COLOMBIANO
CIRCUITOS LOGICOS II
PRIMER EXAMEN PARCIAL (TIPO A)

Nombre: Mauro F. Martinez Quiroga Código: 1820650096

Fecha: 18/04/2020

Indicaciones:

- La duración del examen es 1 hora
- No se permite el uso de apuntes, ni libros, ni hojas con fórmulas
- Solo se recibe el examen en esfero

1. Las interrupciones externas que se ejecutan en el arduino se definen en base a los siguientes pasos
 - a. Pin que recibe el timer, Callback function, Ejecución de rutina.
 - b. Pin señal de disparo, Condición de disparo, Callback function.
 - c. Condición de disparo, Callback function, Periodo de ejecución de la rutina
 - d. Periodo de ejecución de la rutina, No se tiene en cuenta el Pin, Callback function
 - e. Ninguna de las anteriores

RTA: b.

2. Mencione cuales son los puertos que posee el arduino Leonardo, que registros los controlan y escriba la funcionalidad de cada registro

RTA:

- Puertos: B van del B0 al B5.
- Puertos: C van del C0 al C5.
- Puertos: D van del D0 al D7.

Registros: son controlados por 3 registros

- **DDR:** Registro que indica la dirección de la data (Read/Write)
- **PORT:** Registro que indica el Puerto donde va a salir o entrar la data (Read/Write)
- **PIN:** Registro que indica el PIN que va a ser usado (Read Only)

3. Mencione y describa al menos dos métodos que no se ejecutan dentro de una interrupción interna.

RTA:

attachInterrupt (): El primer parámetro para attachInterrupt() es un número de interrupción. Normalmente debe usar digitalPinToInterrupt(pin) para traducir el pin digital real al número de interrupción específico.

detachInterrupt(): Apaga la interrupción dada.

4. Desarrolle un programa por pines, en el que se encienda cada 2 segundos una secuencia diferente de pines.

RTA:

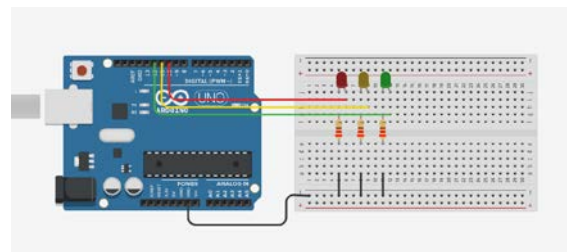
Link Git: <https://github.com/zuyto/Parcial-2-Arquitectura.git>

Código:

```
/******  
/* SEMAFORO usando interrupciones Delay() */  
/******  
  
/*** Mauro Ferney Martines Quiroga ***/  
/*** Profesor: Jonathan Pardo ***/  
/*** se nombran los pines con su respectivo color ***/  
  
int LED_ROJO=10;  
int LED_AMARILLO=11;  
int LED_VERDE=12;  
  
/*** variable para el contador del led verde para intermitencia ***/  
int i=0;  
  
void setup ()  
{  
  /*** se definen los led como pines de salida ***/  
  pinMode(LED_ROJO,OUTPUT);  
  pinMode(LED_AMARILLO,OUTPUT);  
  pinMode(LED_VERDE,OUTPUT);  
}  
  
void loop(){  
  
  /*** Se inicializan los pines en LOW ***/  
  digitalWrite(LED_ROJO,LOW);  
  digitalWrite(LED_AMARILLO,LOW);  
  digitalWrite(LED_VERDE,LOW);  
  delay(10);  
  
  /*** se enciende el led verde durante 2 segundos ***/  
  digitalWrite(LED_VERDE,HIGH);  
  delay(2000);  
  
  i=0;  
  
  /*** PLUS del desarrollo ***/  
  /*** el led verde titila 4 veces con un tiempo de 0.3 segundos encendido y 0.3
```

```
apagado ***/  
while (i<5){  
  digitalWrite(LED_VERDE,LOW);  
  delay(300);  
  digitalWrite(LED_VERDE,HIGH);  
  delay(300);  
  i=i+1;  
}  
  
/*** Apagamos el Led Verde y encendemos el Led Amarillo por 2 segundos ***/  
digitalWrite(LED_VERDE,LOW);  
digitalWrite(LED_AMARILLO,HIGH);  
delay(2000);  
  
/*** apagamos el Led amarillo y encendemos Led Rojo por 2 segundos ***/  
digitalWrite(LED_AMARILLO,LOW); //se apaga el led amarillo  
digitalWrite(LED_ROJO,HIGH); //se enciende el led rojo por 2 segundos  
delay(2000);  
}
```

Simulación:



5. Desarrolle un programa en el cual se registre una comunicación serial entre arduino e interfaz, y en la cual una interrupción externa permite el encendido de un led en configuración por puertos.

RTA:

Link Git: <https://github.com/zuyto/Parcial-2-Arquitectura.git>

Código:

```
int pinButton = 7;
int LED = 12;
int stateLED = LOW;
int stateButton;
int previous = LOW;
long time = 0;
long debounce = 200;
```

```
void setup() {
  pinMode(pinButton, INPUT);
  pinMode(LED, OUTPUT);
}
```

```
void loop() {
  stateButton = digitalRead(pinButton);
  if(stateButton == HIGH && previous == LOW && millis() - time > debounce) {
    //if button is pressed and LED is off
    if(stateLED == HIGH){
      stateLED = LOW;
    } else {
      stateLED = HIGH;
    }
  }
  time = millis();
}
```

```
}
digitalWrite(LED, stateLED);
previous == stateButton;
}
```

Simulación:

