

Arquitectura de Computadoras 1
Licenciatura en Ciencias de la Computación
Universidad Nacional de Cuyo
Ejemplos de programas con Arduino IDE

Ejemplo 1:

El siguiente programa hace parpadear el led conectado al pin 13. El led se enciende un segundo, luego se apaga un segundo, y repite el ciclo infinitamente. No incluye interrupciones ni funciones.

```
void setup()
{
  pinMode(13, OUTPUT); //Configura el pin 13 como salida.
}

void loop()
{
  digitalWrite(13, HIGH); //Enciende el led conectado al pin 13.
  delay(1000); //Espera 1000 milisegundos = 1 segundo.
  digitalWrite(13, LOW); //Apaga el led conectado al pin 13.
  delay(1000); //Espera 1000 milisegundos = 1 segundo.
}
```

Ejemplo 2:

El siguiente programa hace parpadear el led conectado al pin 13. Al apretar el pulsador 2, el parpadeo acelera. Al apretar el pulsador 3, el parpadeo vuelve a su velocidad inicial. Incluye interrupciones. No incluye llamadas a funciones.

```
int demora=1000; /*Define la variable global demora. Le da
valor inicial de 1000*/
void setup()
{
  pinMode(13, OUTPUT); //Configura el pin 13 como salida.
  pinMode(2, INPUT); //Configura el pin 2 como entrada.
  pinMode(3, INPUT); //Configura el pin 3 como entrada.
  attachInterrupt(0,pin2_ISR,RISING); /*Activa la interrupción
del pin 2 (interrupción 0). Indica que la función llamada
pin2_ISR tiene el código de la rutina de servicio. Indica que el
evento que debe disparar la interrupción será un flanco de
subida sobre el pin 2 */
  attachInterrupt(1,pin3_ISR,RISING); /*Activa la interrupción
```

```

del pin 3 (interrupción 1). Indica que la función llamada
pin3_ISR tiene el código de la rutina de servicio. Indica que el
evento que debe disparar la interrupción será un flanco de
subida sobre el pin 3 */
}
void loop()
{
digitalWrite(13, HIGH); //Enciende el led conectado al pin 13.
delay(demora); /*Espera los milisegundos indicados por la
variable demora.*/
digitalWrite(13, LOW); //Apaga el led conectado al pin 13.
delay(demora); /*Espera los milisegundos indicados por la
variable demora.*/
}

void pin2_ISR(){
demora=100; //Disminuye el valor de la variable demora a 100.
}

void pin3_ISR(){
demora=1000; //Hace que la variable demora vuelva a valer 1000.
}

```

Ejemplo 3:

El siguiente programa hace que los led conectados a los pines desde el 6 al 13 se enciendan de izquierda a derecha y luego de derecha a izquierda. El ciclo se repite infinitamente. Al pulsar el pulsador conectado al pin 2, la simulación va más rápido (se divide la variable demora por 2). Al pulsar el pulsador conectado al pin 3, la velocidad de la simulación vuelve al valor inicial (demora=1000). Se utilizan interrupciones y llamadas a funciones (notar la diferencia entre interrupciones y llamadas a funciones).

```

int demora=1000; /*Define la variable global demora. Le da
valor inicial de 1000*/
void setup()
{
for(int i=6;i<=13;i=i+1){
pinMode(i, OUTPUT); //Configura el pin i como salida.
}
pinMode(2, INPUT); //Configura el pin 2 como entrada.
pinMode(3, INPUT); //Configura el pin 3 como entrada.

```

```

attachInterrupt(0,pin2_ISR,RISING); /*Activa la interrupción
del pin 2 (interrupción 0). Indica que la función llamada
pin2_ISR tiene el código de la rutina de servicio. Indica que el
evento que debe disparar la interrupción será un flanco de
subida sobre el pin 2 */
attachInterrupt(1,pin3_ISR,RISING); /*Activa la interrupción
del pin 3 (interrupción 1). Indica que la función llamada
pin3_ISR tiene el código de la rutina de servicio. Indica que el
evento que debe disparar la interrupción será un flanco de
subida sobre el pin 3 */
}

void loop()
{
encender_hacia_la_derecha(); //Llamada a función
encender_hacia_la_izquierda(); //Llamada a función
}

void encender_hacia_la_derecha(){
for(int i=13;i>=7;i=i-1){
digitalWrite(i, HIGH); //Enciende el led conectado al pin i.
delay(demora); /*Espera los milisegundos indicados por demora.*/
digitalWrite(i, LOW); //Apaga el led conectado al pin i.
}
}

void encender_hacia_la_izquierda(){
for(int i=6;i<=12;i=i+1){
digitalWrite(i, HIGH); //Enciende el led conectado al pin i.
delay(demora); /*Espera los milisegundos indicados por demora.*/
digitalWrite(i, LOW); //Apaga el led conectado al pin i.
}
}

void pin2_ISR(){
demora=demora/2; /*Disminuye el valor de la variable demora a la
mitad*/
}

void pin3_ISR(){
demora=1000; //Hace que la variable demora vuelva a valer 1000.
}

```

Ejemplo 4:

El siguiente programa utiliza el sensor ultrasónico de distancia. Mide distancias cada segundo. Al presionar el pulsador 2, el medidor se enciende. Al presionar el pulsador 3, la lectura se apaga. Utiliza interrupciones, llamadas a funciones y el puerto serial. Para ver lo que se imprime en el puerto serial, debe hacer clic en "Serial Monitor"

```
/*Variables globales.
Nota: dar nombres a los pines hace los programas más fáciles de leer*/
int trig=5; //Se usará para nombrar el pin trig.
int echo=4; //Se usará para nombrar el pin echo.
bool encendido=true; /*Se usará para indicar si la lectura está
encendida o apagada.*/
void setup()
{
  pinMode(2, INPUT); //Configura el pin 2 como entrada.
  pinMode(3, INPUT); //Configura el pin 3 como entrada.
  pinMode(6, OUTPUT); //Configura el pin 6 como salida.
  pinMode(echo, INPUT); //Configura el pin echo (pin 4) como entrada.
  pinMode(trig, OUTPUT); //Configura el pin trig (pin 5) como salida.
  attachInterrupt(0,pin2_ISR,RISING); /*Activa la interrupción
del pin 2 (interrupción 0). Indica que la función llamada
pin2_ISR tiene el código de la rutina de servicio. Indica que el
evento que debe disparar la interrupción será un flanco de
subida sobre el pin 2 */
  attachInterrupt(1,pin3_ISR,RISING); /*Activa la interrupción
del pin 3 (interrupción 1). Indica que la función llamada
pin3_ISR tiene el código de la rutina de servicio. Indica que el
evento que debe disparar la interrupción será un flanco de
subida sobre el pin 3 */
  Serial.begin(9600); /*Inicializa el puerto serial. Para ver lo que
se escribe en el puerto serial, debe hacer clic en "Serial Monitor"*/
}

void loop()
{
  if(encendido==true){ //Bloque a ejecutar si el medidor está encendido
    generar_pulso_trig(); //Función que genera el pulso en el pin trig.
    leer_distancia(); //Función que lee la distancia.
    digitalWrite(6,HIGH); /*Las siguientes 4 instrucciones hacen
parpadear el led 6 por cada lectura*/
    delay(500);
  }
}
```

```

    digitalWrite(6,LOW);
    delay(500);
}
else{ //Bloque a ejecutar si el medidor está apagado
    Serial.println("Distancia: Medidor apagado"); /* Imprime en el
puerto serie.*/
    delay(1000);
}
}

void generar_pulso_trig(){
    /*Esta función genera un pulso en el pin trig. Un pulso es poner el
pin en bajo, luego en alto, y luego nuevamente en bajo*/
    digitalWrite(trig, LOW);
    delayMicroseconds(20);
    digitalWrite(trig, HIGH); //Pone en alto el pin trig (pin 5).
    delayMicroseconds(20); /*Espera 20 microsegundos*/
    digitalWrite(trig, LOW); //Pone en bajo el pin trig (pin 5).
}

void leer_distancia(){
    /*Función que lee el tiempo que demora */
    unsigned long distancia;
    distancia=pulseIn(echo,1,1000000)/58;
    Serial.println("Distancia: " + String(distancia) + " cm"); /*Imprime
en el puerto serie el valor de la distancia. Notar que no se puede
concatenar números enteros con cadenas de caracteres. Debe
transformarse los Enteros a cadena de caracteres.*/
}

void pin2_ISR(){
    encendido=true; //Enciende la lectura
}

void pin3_ISR(){
    encendido=false; //Apaga la lectura
}

```