

Práctica 9 - Programación del LED integrado

Medina Medina, David A.
Benlliure Jiménez, M^o Cristina
Brito Ramos, Christian
López González, Néstor

22 de abril de 2019

Índice general

| | |
|--|----------|
| 1. Introducción | 2 |
| 2. Método y materiales | 3 |
| 2.1. Materiales | 3 |
| 2.2. Método | 3 |
| 2.2.1. Macros y variables globales | 3 |
| 2.2.2. Rutina <code>setup()</code> | 4 |
| 2.2.3. Rutina <code>loop()</code> | 4 |
| 3. Conclusiones | 6 |
| Referencias | 7 |

Introducción

Arduino es el primer proyecto de hardware en open-source que facilita el diseño de productos digitales que requieren de un microcontrolador.

El objetivo de este proyecto (Benlliure Jiménez, Medina Medina, Brito Ramos, y López González, s.f.) consiste en utilizar la placa *Arduino UNO R3* para controlar la frecuencia de parpadeo del LED integrado en la placa, de modo que, la frecuencia de parpadeo sea mayor cuanto más positivo sea el valor de la onda portadora senoidal.

Método y materiales

2.1. Materiales

El desarrollo de este proyecto se ha llevado a cabo utilizando el IDE de desarrollo de aplicaciones *C/C++* de *JetBrains*, *CLion*, y las siguientes herramientas:

- Placa Arduino Uno R3.
- Librería `Arduino.h` (Arduino, s.f.).
- Archivo `CMakeLists.txt` para la construcción automatizada de las dependencias de la librería `Arduino.h`.

El proyecto también puede cargarse en la placa usando el el IDE oficial de *Arduino*.

2.2. Método

El código que se cargará en la placa se encuentra en el archivo `Practica9.ino` el cual se estructura en las siguientes partes:

- Macros y variables globales
- Rutina `setup()`.
- Rutina `loop()`.

2.2.1. Macros y variables globales

En primer lugar, se incluye la librería `Arduino.h` que contiene todas las rutinas primitivas que utilizaremos para comunicarnos con la placa.

Se declaran un conjunto de macros que serán utilizadas en el código:

AMPLITUDE Esta macro contiene el valor de la amplitud de la onda sinusoidal que generaremos. Por defecto, su valor es igual a 1 unidad.

MIN_FREQ Se establece el valor de la frecuencia de la onda senoidal. Por defecto: 1 Hz.

Rutina `sine_wave(f, t)` Esta macro define una rutina que tomará como parámetro de entrada una frecuencia, `f`, y una magnitud de tiempo, `t`. La onda senoidal generada esta definida por esta ecuación,

$$s(f, t) = A \sin(2\pi ft)$$

donde A es la amplitud de onda, f es la frecuencia y t el tiempo.

En último lugar se inicializa una variable global a 0 (`clockTick`). Esta variable incrementa su valor en una unidad en cada iteración de la rutina `loop()` para poder extraer el valor de la onda senoidal generada en un tick de tiempo determinado.

```
1 #include <Arduino.h>
2
3 #define AMPLITUDE          1.
4 #define MIN_FREQ          1.
5 #define sine_wave(f, t)    ( AMPLITUDE * sin(2 * PI * f * t) )
6
7 double clockTick = 0.f;
```

2.2.2. Rutina `setup()`

Esta primitiva realiza los ajustes iniciales de la placa. Se establece la velocidad de transmisión de datos de la placa a 9600 baudios y se configura el pin número 13 (definido por la macro primitiva `LED_BUILTIN`) como pin de salida.

```
1 void setup() {
2     Serial.begin(9600);
3     pinMode(LED_BUILTIN, OUTPUT);
4 }
```

2.2.3. Rutina `loop()`

Esta rutina primitiva define el bucle principal que se ejecuta en la placa *Arduino*.

En la variable `s` se guarda el valor de la función sinusoidal con frecuencia mínima, `FREQ_MIN`, y un tiempo, `clockTick`.

La frecuencia de parpadeo se especifica en la variable `f`, donde la frecuencia variará su valor en el rango $[MIN_FREQ, 2 \times MIN_FREQ]$ determinado por la siguiente fórmula,

$$f = MIN_FREQ + MIN_FREQ(s/AMPLITUDE) + MIN_FREQ$$

donde MIN_FREQ es la frecuencia mínima de oscilación del LED –coincide con la frecuencia de la onda senoidal generada–, s es el valor de la onda sinusoidal para un tick de tiempo determinado y $AMPLITUDE$ es la amplitud máxima de la onda generada.

A continuación, se define en la variable `t` el periodo en segundos de parpadeo del LED con la siguiente ecuación,

$$t = 1/f$$

donde t es el periodo y f la frecuencia de parpadeo.

Este periodo se a milisegundos multiplicando por 1000 y se divide entre dos con el objeto de encender el LED en la primera mitad del periodo y apagarlo en la segunda mitad.

Finalmente, se avanza el tick de reloj $\frac{1}{16}$ unidades de tiempo. Si al incrementar el tick, este supera el valor de 1000 se cicla su valor a 0.

```
1 void loop() {
2     double s = sine_wave(MIN_FREQ, clockTick);
3     double f = MIN_FREQ + MIN_FREQ * (s / AMPLITUDE) + MIN_FREQ;
4
5     double t = 1/f;
6
7     delay(t/2 * 1000);
8     digitalWrite(LED_BUILTIN, HIGH);
9     delay(t/2 * 1000);
10    digitalWrite(LED_BUILTIN, LOW);
11
12    clockTick = clockTick + 1.f/16.f;
13    if (clockTick >= 1000.f) clockTick = 0.f;
14 }
```

Conclusiones

Tras cargar el código descrito en este documento, el LED integrado de la placa *Arduino Uno R3* se comporta de la manera esperada, de modo que, cuando el valor de la onda senoidal es máximo, la frecuencia de parpadeo del LED se duplica ($2 \times MIN_FREQ$) y cuando su valor es mínimo su frecuencia es exactamente igual a la frecuencia de oscilación de la onda senoidal generada (MIN_FREQ).

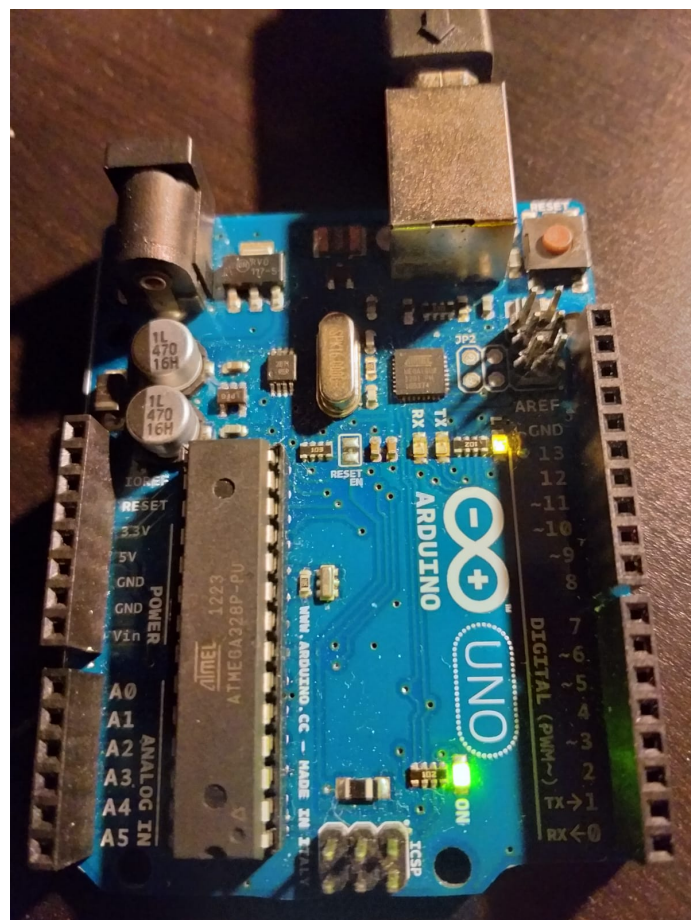


Figura 3.1: LED integrado de la placa *Arduino* parpadeando

Referencias

- Arduino. (s.f.). *Language Reference*. <https://www.arduino.cc/reference/en/>. Autor. (Accessed: 2019-04-16)
- Benlliure Jiménez, M. C., Medina Medina, D. A., Brito Ramos, C., y López González, N. (s.f.). *Repositorio CIU - Practica 9*. <https://github.com/david00medina/CIU-Repositorio/tree/master/Practica9>. Medina Medina, David A. (Accessed: 2019-04-16)