
CEFET-MG

Engenharia da Computação

Disciplina: Robótica e Controle Digital de Sistemas Dinâmicos

Professor: Ramon C. Lopes

AULA PRÁTICA 01

Assunto: Conversor R2R

Objetivo: Verificar experimentalmente a conversão de números binários em números decimais (Conversor D/A)

1 Introdução

A conversão de números da base decimal para binária (valores analógicos para digitais) é vista na figura a seguir:

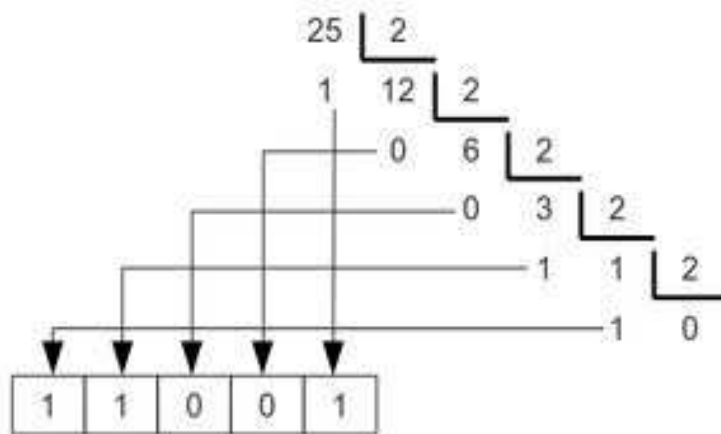


Figura 1: Conversão D/A

A conversão de números da base binária para decimal (valores digitais para analógicos) é vista na figura abaixo:

$$\begin{array}{c} 100011_{(2)} = 35_{(10)} \\ \swarrow \quad \downarrow \quad \searrow \quad \swarrow \quad \searrow \quad \swarrow \\ 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ 32 + 0 + 0 + 0 + 2 + 1 = 35_{(10)} \end{array}$$

Figura 2: Conversão D/A

Um circuito denominado R2R pode converter valores digitais em analógicos, como será apresentado neste guia.

2 Atividade prática 1

Digite o código abaixo na interface do arduino para verificar o funcionamento:

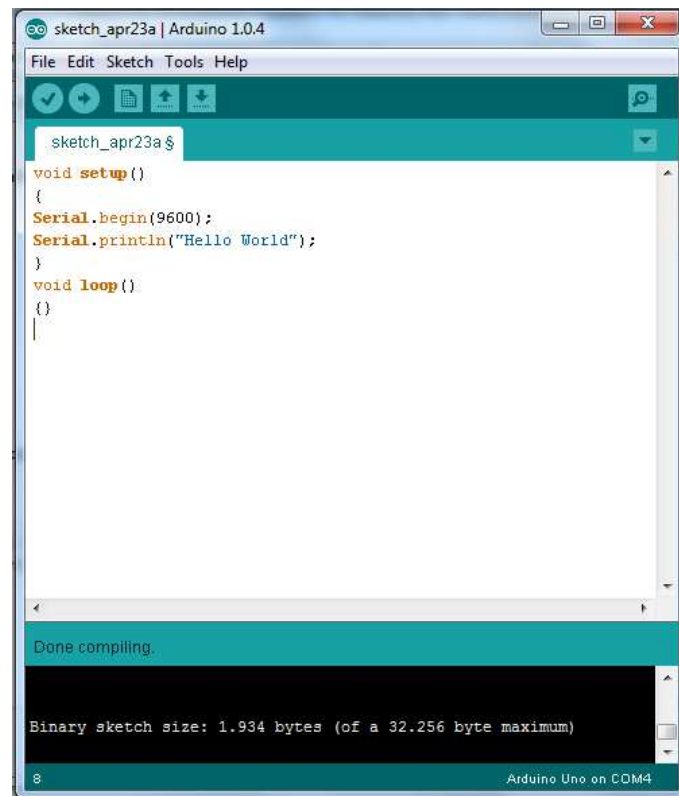


Figura 3: Código inicial no arduino

Utilize os comandos, Verify, Upload e Serial Monitor disponíveis na tela da interface do arduino para obter a resposta a seguir:

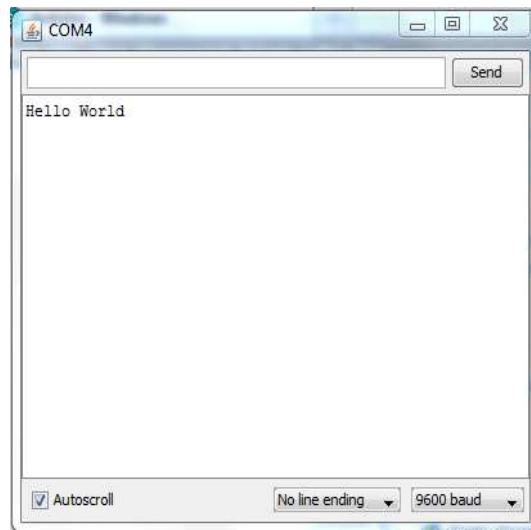


Figura 4: Código inicial no arduino

3 Atividade prática 2 - Entrada Analógica

Implemente o código abaixo no arduino:

Code

```
/*
 * AnalogInput
 * by DojoDave <http://www.0j0.org>
 *
 * Turns on and off a light emitting diode(LED) connected to digital
 * pin 13. The amount of time the LED will be on and off depends on
 * the value obtained by analogRead(). In the easiest case we connect
 * a potentiometer to analog pin 2.
 */
int potPin = 2; // select the input pin for the potentiometer
int ledPin = 13; // select the pin for the LED
int val = 0; // variable to store the value coming from the sensor
void setup() {
  pinMode(ledPin, OUTPUT); // declare the ledPin as an OUTPUT
}
void loop() {
  val = analogRead(potPin); // read the value from the sensor
  digitalWrite(ledPin, HIGH); // turn the ledPin on
  delay(val); // stop the program for some time
  digitalWrite(ledPin, LOW); // turn the ledPin off
  delay(val); // stop the program for some time
}
```

Figura 5: Código para implementar entrada analógica

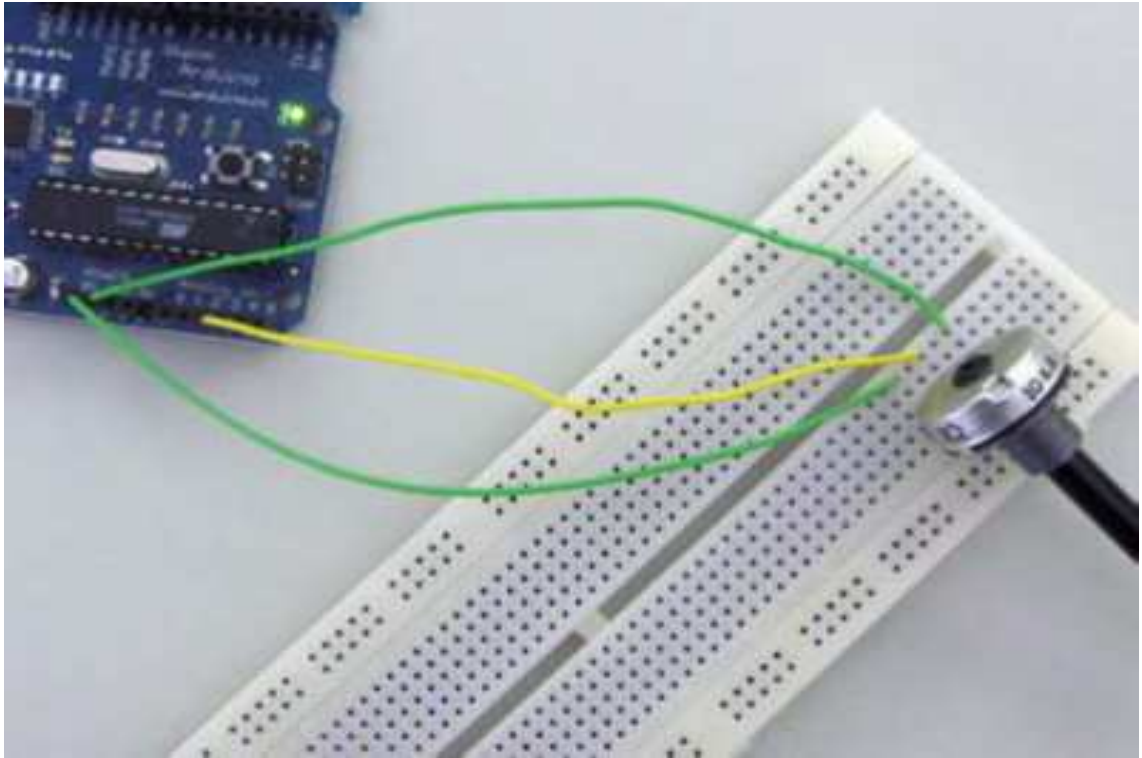


Figura 6: Circuito para implementar entrada analógica

4 Atividade prática 3 - Saída digital

Implemente o código abaixo no arduino:

Code

```
int timer = 100; // The higher the number, the slower the timing.
int pins[] = { 2, 3, 4, 5, 6, 7 }; // an array of pin numbers
int num_pins = 6; // the number of pins (i.e. the length of the array)
void setup()
{
  int i;
  for (i = 0; i < num_pins; i++) // the array elements are numbered from 0 to num_pins - 1
    pinMode(pins[i], OUTPUT); // set each pin as an output
}
void loop()
{
  int i;
  for (i = 0; i < num_pins; i++) { // loop through each pin...
    digitalWrite(pins[i], HIGH); // turning it on,
    delay(timer); // pausing,
    digitalWrite(pins[i], LOW); // and turning it off.
  }
  for (i = num_pins - 1; i >= 0; i--) {
    digitalWrite(pins[i], HIGH);
    delay(timer);
    digitalWrite(pins[i], LOW);
  }
}
```

Figura 7: Código para implementar saída digital

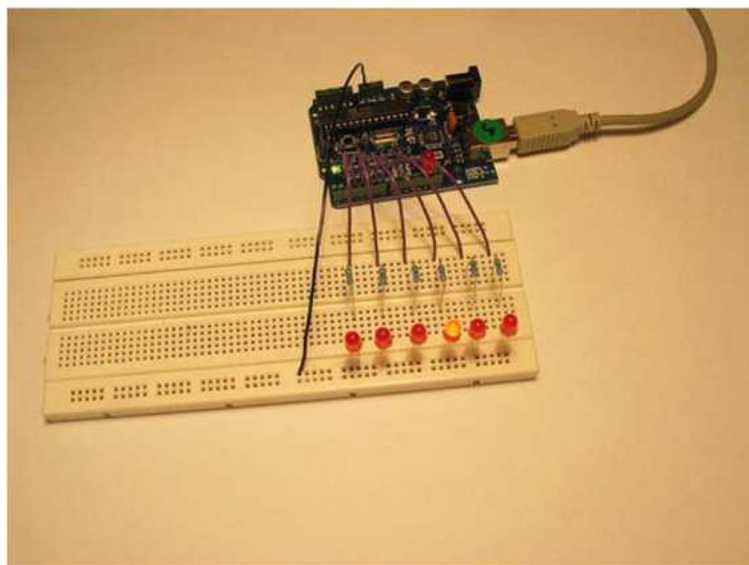


Figura 8: Circuito para implementar saída digital

5 Atividade prática 4 - Saída analógica

Elabore um programa que leia a entrada analógica, filtre através da equação de diferenças e converta em números binários na saída para um conversor R2R. Elabore um filtro digital passa-baixas a partir da equação de diferenças $y^k = .1 * v^k + .9048 * v^{k-1}$ extraída da função de transferência $H(f) = \frac{.1}{1 + .9048z^{-1}}$.

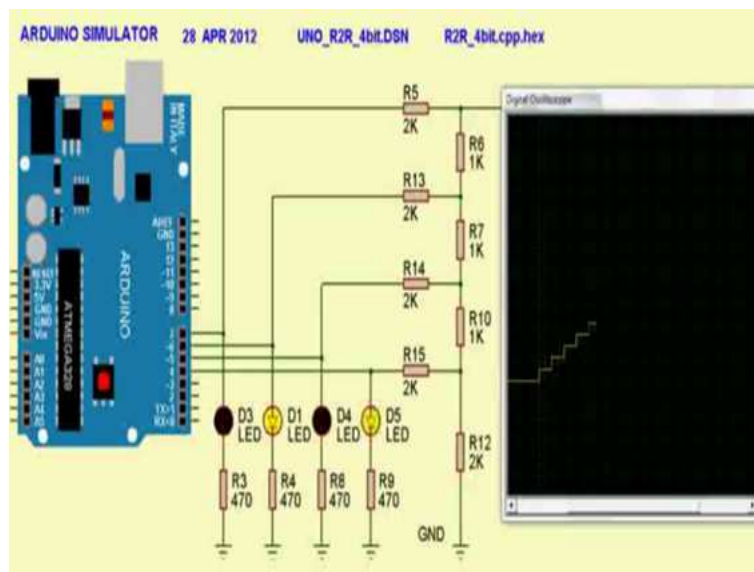


Figura 9: Circuito para implementar saída analógica

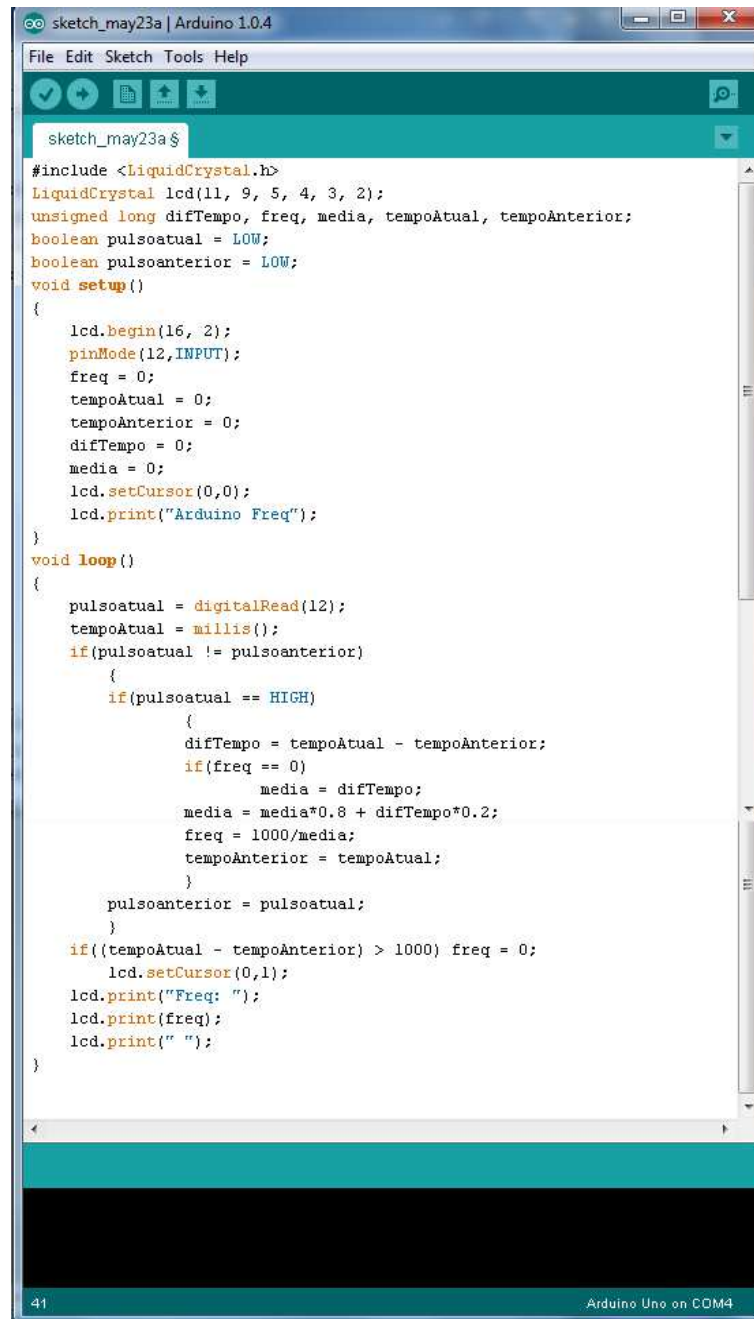
Code

```
/*Digital-to-Analog converter using resistors in an R-2R array*/
void setup()
{
  // initialize the digital pins 4,5,6 and 7 as outputs.
  pinMode(2, OUTPUT); // set each pin as an output
  pinMode(3, OUTPUT); // set each pin as an output
  pinMode(4, OUTPUT); // set each pin as an output
  pinMode(5, OUTPUT); // set each pin as an output
  pinMode(6, OUTPUT); // set each pin as an output
  pinMode(7, OUTPUT); // set each pin as an output
}
void loop()
{
  digitalWrite(2, HIGH); // turning it on,
  delay(100); // pausing,
  digitalWrite(2, LOW); // and turning it off.
  digitalWrite(3, HIGH); // turning it on,
  delay(100); // pausing,
  digitalWrite(3, LOW); // and turning it off.
  digitalWrite(4, HIGH); // turning it on,
  delay(100); // pausing,
  digitalWrite(4, LOW); // and turning it off.
  digitalWrite(5, HIGH); // turning it on,
  delay(100); // pausing,
  digitalWrite(5, LOW); // and turning it off.
  digitalWrite(6, HIGH); // turning it on,
  delay(100); // pausing,
  digitalWrite(6, LOW); // and turning it off.
  digitalWrite(7, HIGH); // turning it on,
  delay(100); // pausing,
  digitalWrite(7, LOW); // and turning it off.
}
```

Figura 10: Código para implementar saída analógica

6 Atividade prática 5 - Frequencímetro digital

Obtenha um frequencímetro através do código da figura 11:



```
sketch_may23a $
#include <LiquidCrystal.h>
LiquidCrystal lcd(11, 9, 5, 4, 3, 2);
unsigned long difTempo, freq, media, tempoAtual, tempoAnterior;
boolean pulsoatual = LOW;
boolean pulsoanterior = LOW;
void setup()
{
  lcd.begin(16, 2);
  pinMode(12, INPUT);
  freq = 0;
  tempoAtual = 0;
  tempoAnterior = 0;
  difTempo = 0;
  media = 0;
  lcd.setCursor(0,0);
  lcd.print("Arduino Freq");
}
void loop()
{
  pulsoatual = digitalRead(12);
  tempoAtual = millis();
  if(pulsoatual != pulsoanterior)
  {
    if(pulsoatual == HIGH)
    {
      difTempo = tempoAtual - tempoAnterior;
      if(freq == 0)
      {
        media = difTempo;
        media = media*0.8 + difTempo*0.2;
        freq = 1000/media;
        tempoAnterior = tempoAtual;
      }
      pulsoanterior = pulsoatual;
    }
    if((tempoAtual - tempoAnterior) > 1000) freq = 0;
    lcd.setCursor(0,1);
    lcd.print("Freq: ");
    lcd.print(freq);
    lcd.print(" ");
  }
}
```

41 Arduino Uno on COM4

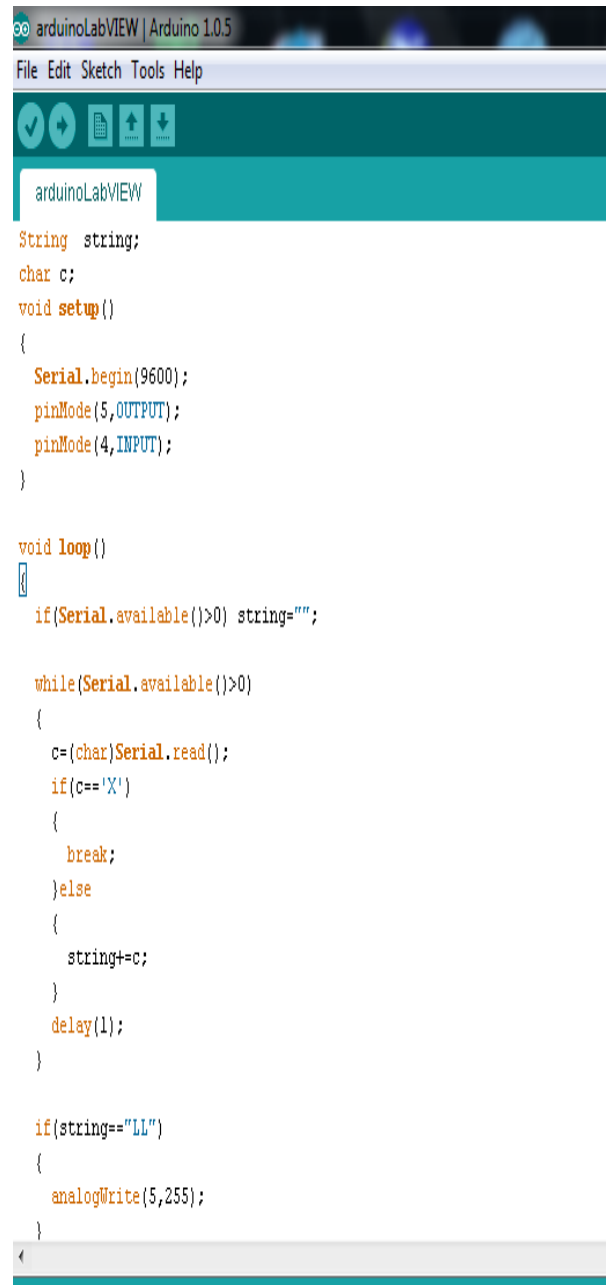
Figura 11: Código para implementar o frequencímetro digital

7 Atividade prática 6 - Enviando dados do LabVIEW para o arduino

<https://www.youtube.com/watch?v=AddWr5cD-d0>

8 Atividade prática 7 - Enviando dados do arduino para o LabVIEW

<https://www.youtube.com/watch?v=wLovmFfe5GM>



The screenshot shows the ArduinoLabVIEW IDE window titled "arduinoLabVIEW | Arduino 1.0.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for a checkmark, a circular arrow, a document, an upload arrow, and a download arrow. A tab labeled "arduinoLabVIEW" is active. The code editor displays the following C++ code:

```
String string;
char c;
void setup()
{
  Serial.begin(9600);
  pinMode(5,OUTPUT);
  pinMode(4,INPUT);
}

void loop()
{
  if(Serial.available()>0) string="";

  while(Serial.available()>0)
  {
    c=(char)Serial.read();
    if(c=='X')
    {
      break;
    }else
    {
      string+=c;
    }
    delay(1);
  }

  if(string=="LL")
  {
    analogWrite(5,255);
  }
```

