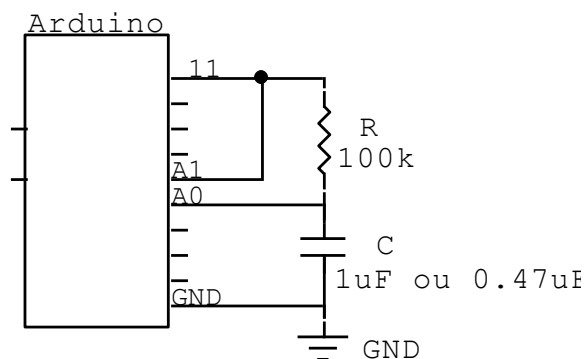


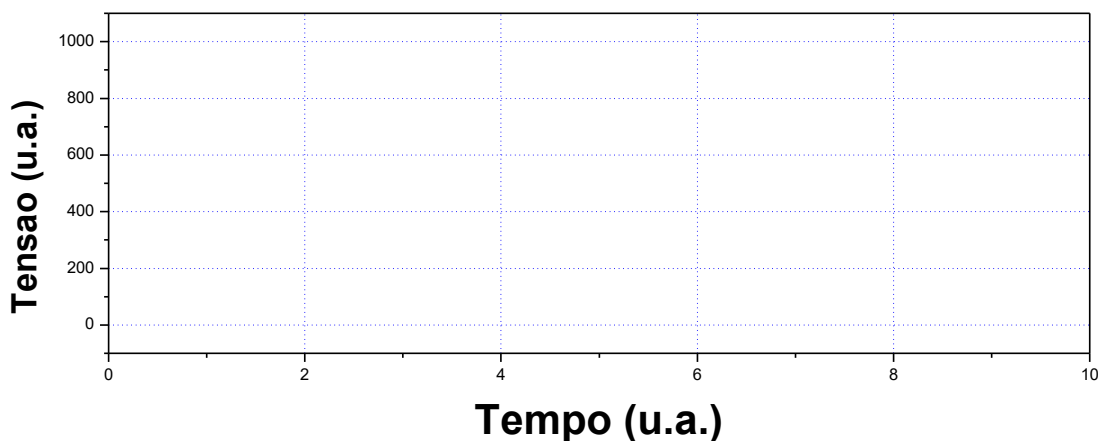
Prática 2 – Circuito RC.

- 1) Monte no proto-board o circuito da figura abaixo, usando o resistor de $100\text{ k}\Omega$ e capacitor de $0,47\text{ }\mu\text{F}$ num primeiro momento e o de $1\text{ }\mu\text{F}$ na sequência. (Você pode tentar também com os dois capacitores em paralelo.)



Esse é um circuito típico de carga e descarga de capacitores. Como será discutido junto com o programa do Arduino, o pino 11 das Entradas/Saídas Digitais estabelecerá 5 V ou 0 V no resistor, promovendo a carga ou descarga do capacitor.

- 2) Para avaliar o funcionamento, isto é, fazer uma observação do processo de carga e descarga do capacitor usa-se ferramenta “Plotter Serial” da interface do Arduino. Repare que a porta serial do Arduino foi configurada para a velocidade de 19200 bps. Após abrir a Ferramenta “Plotter Serial”, no canto embaixo à esquerda ajuste a velocidade para 19200. Se tudo estiver certo, será apresentado na tela um gráfico da tensão da entrada (pino A1) e da tensão sobre o capacitor (pino A0) em função do tempo. Represente no gráfico abaixo (ou anexe um print screen) o que você observou, indicando qual dos traços é a tensão de entrada e sobre o capacitor. (Perceba que tanto o eixo do tempo como o das tensões estão com escalas arbitrárias, faça o gráfico de tal maneira que 3 ou 4 ciclos do seu traçado ocupe o gráfico inteiro).



- 3) A fim de fazer a medida da constante de tempo RC do circuito usa-se a ferramenta “Monitor Serial”. Nesse caso será apresentado na tela uma lista com 3 colunas; a 1ª indica o instante de tempo em que as medidas foram feitas, a 2ª o valor lido sobre o capacitor e a 3ª o valor lido na entrada do circuito. Copie o conteúdo das colunas durante um processo de descarga e transfira para algum editor de gráficos. Faça uma linearização dos dados e (a) Preencha a tabela abaixo e (b) diga porque, apesar de o processo de carga também ser do tipo exponencial, ele (diretamente) não se presta para fazermos uma linearização do gráfico e obtermos a constante de tempo RC. Para facilitar a construção do gráfico, os dados da coluna tempo reiniciam sempre que começa a carga ou descarga do capacitor. (c) Perceba que os valores lidos na entrada e no capacitor não estão em volts. A constante de tempo determinada nesse exercício mudaria se tais valores fossem medidos em volts?

	RC calculado (s)	RC medido (s)
C = 0,47 μ F		
C = 1 μ F		

- 4) Como você alteraria o sketch do Arduino para obter também informação sobre a corrente no capacitor? (Apenas indique o que deveria ser feito, não é necessário alterar o programa. Mas se quiseres alterar, óbvio, fique a vontade. Eu ficaria feliz com sua nova versão.)



```
//Define as variáveis
int pinoSaida = 11;
float tempoInicial;
float tempoFinal;
float tempoTotal;
double resistencia;
int pinoCapacitor;
int pinoEntrada;
.....
```

Aqui estão as definições iniciais das variáveis que serão usadas na programação do Arduino.

```
void setup() {
// Configura o pino para saída
pinMode(pinoSaida, OUTPUT);
digitalWrite(pinoSaida,LOW);
```

Configura o pino 11 das entradas/saídas digitais como sendo saída (OUTPUT) e atribui a ele o valor 0 V, (LOW).

```
// Configura a velocidade da serial
Serial.begin(19200);
}
```

Define a velocidade da porta serial como sendo 19200 bps

```
void loop() {
// Coloca o pino em estado para descarregar o capacitor
digitalWrite(pinoSaida,LOW);
//Marca tempo inicial
tempoInicial=micros();
//Lê a tensão no capacitor
pinoCapacitor=analogRead (A0);
```

Aqui começa o loop de medida

Começa colocando o pino 11 em 0V e estabelece o tempo 0s. A partir daqui o tempo começa a ser medido.

```
// Continua a descarga do capacitor enquanto a su
while(pinoCapacitor > 10) {
pinoCapacitor=analogRead (A0);
//Lê a tensão de entrada
pinoEntrada=analogRead (A1);
// Marca tempo Final
tempoFinal = micros();
//Calcula o tempo total (constante de tempo)
tempoTotal = tempoFinal-tempoInicial;
Serial.print(tempoTotal/1000000);
Serial.print(" ");
Serial.print(pinoCapacitor);
Serial.print(" ");
Serial.println(pinoEntrada);
```

Dentro dessa rotina “while”, enquanto a entrada analógica A0, que lê a ddp sobre o capacitor for menos do que 10, correspondendo a 0,05 V, o Arduino ficará lendo as entradas A0, A1 e marcando o tempo em que as medidas foram feitas. Além disso, ele escreve na porta serial esses valores para serem lidos por nós ou pela ferramenta “Monitor Serial” ou pela “Ploter Serial”.

```

//Coloca o pino em estado alto para carregar o capacitor
digitalWrite(pinoSaida, HIGH);
//Marca tempo inicial
tempoInicial=micros();
//Continua a carga do capacitor enquanto sua tensão for menor do que 1023
while (pinoCapacitor < 1013) { // Se 1023 corre:
pinoCapacitor = analogRead (A0);
//Lê a tensão de entrada
pinoEntrada=analogRead (A1);
// Marca tempo Final
tempoFinal = micros();
//Calcula o tempo total
tempoTotal = tempoFinal-tempoInicial;
Serial.print(tempoTotal/1000000);
Serial.print("    ");
Serial.print(pinoCapacitor);
Serial.print("    ");
Serial.println(pinoEntrada);
}

```

Na rotina anterior quando a leitura de A0 baixar de 10, o programa vem para essa rotina, começa colocando o pino 11 em +5V para o processo de carga do capacitor e fará o mesmo que antes, ou seja, enquanto a ddp no capacitor for menor do que 1013 ele lerá as entradas analógicas A0 e A1, anotará o instante de tempo em que as medidas foram feitas e escreverá essas informações na porta serial.

Quando A0 for maior do que 1013 ele voltará para a rotina do processo de descarga.