



Arduino UNO R3: Seno

Física Aplicada à Computação - Licenciatura em Engenharia Informática

Hugo Alexandre Silva

Beja - Portugal

Sumário

Trabalho Laboratorial: Arduino UNO R3: Seno	1
Introdução	3
Objetivo	5
Aparato Experimental	5
Sensor de temperatura	6
Breadboard	7
Led	8
Resistores	9
Seno	10
Código parteum	15
Seno de 7	18
Media final MCM	22
Media final MMM	24
Circuito e esquemático em fritzing	25
GNUPLOT	31
TESTES DE EXECUÇÃO	33
Conclusão	39
Referências	40

Introdução

Este trabalho visa executar tarefas em Arduino que é uma plataforma de prototipagem eletrônica de hardware em placa única, projetada inicialmente com um microcontrolador Atmel AVR, trabalhando com periféricos com conexões de entrada e saída constituído por 14 pinos, assim como circuitos eletrônicos e manipuláveis em uma linguagem de programação padrão. A linguagem tem origem em Wiring e é essencialmente C/C++. A visão do projeto é criar ferramentas acessíveis, com baixo custo, flexíveis e fáceis de se usar por principiantes e profissionais, aqueles que não teriam alcance aos controladores mais sofisticados e ferramentas mais complicadas. Em relação a pinagem são: 6 podem ser usados como saídas PWM, 6 entradas analógicas, uma conexão USB, uma entrada de alimentação com conexão ICSP e um botão de reset. Cada um dos 14 pinos digitais do UNO podem ser utilizados como uma entrada ou saída utilizando-se as funções `pinMode()`, `digitalWrite()` e `digitalRead()`. Eles operam a 5.0V e cada pino pode fornecer ou receber um máximo de 40mA.

Neste trabalho utilizamos o Arduino UNO R3 como sistema de aquisição de dados de sensores nomeadamente sensores da temperatura com cálculos e formulas de seno, temperatura, ruído, somatório de série de 7 senos, as quais são: $T=f(\sin(wt))+A*\text{random}(R_{\min},R_{\max})$, $T=f(\sum_{k=1}^N \sin(kwt))+A*\text{random}(R_{\min},R_{\max})$. Posteriormente, o tratamento do código com o intuito de desenvolver competências de programação de sistema de dados e tratamento elementar.

Ao Arduino UNO R3 podemos complementar vários sensores como, por exemplo, sensores ambientais, ópticos, força e flexão, som e temperatura e humidade relativa.

O presente trabalho estruturar-se-á do seguinte modo:

“Primeiro será uma variação sinusoidal entre valores arbitrários T_{\min} e T_{\max} , definidos como constantes, com ruído obtido a partir do gerador de números pseudo aleatórios, `random()`, de acordo com a formula citada acima.

Definição de dois níveis de alerta T_0 e T_1 correspondentes às seguintes situações: ° Se $T_{min} < T \leq T_0$ mensagem “LED Green” / acende LED verde ° Se $T_0 < T \leq T_1$ mensagem “LED Green+Red” / acende LED verde e vermelho ° Se $T_1 < T \leq T_{max}$ mensagem “LED Red” / acende LED vermelho.

Segundo, será uma versão do programa o sinal deverá ser gerado não apenas por uma função seno, mas através de uma soma de N funções de acordo com a citada acima. O parâmetro N , que define o número de termos do somatório, é definido pelo utilizador. A figura 5 mostra um exemplo do sinal gerado a partir de uma série de 7 senos.

Média corrente cumulativa (MCM): $T_{med} = (T_1 + T_2 + \dots + T_k) / k$, em que k é o número de medições realizadas; • Média corrente de M medições (MMM): $T_M = (T_k + T_{k+1} + \dots + T_{k+M-1}) / M$, em que M é número medições que entram no cálculo; • Smoothing: $T_s = T_k + (T_{k+1} - T_k) \times f_s$, em que T_s é a medição suavizada calculada a partir de duas medições consecutivas, e f_s o factor de suavização ($0 < f_s \leq 1$)”.

Citação do manual projeto FAC – Mestre Nuno.

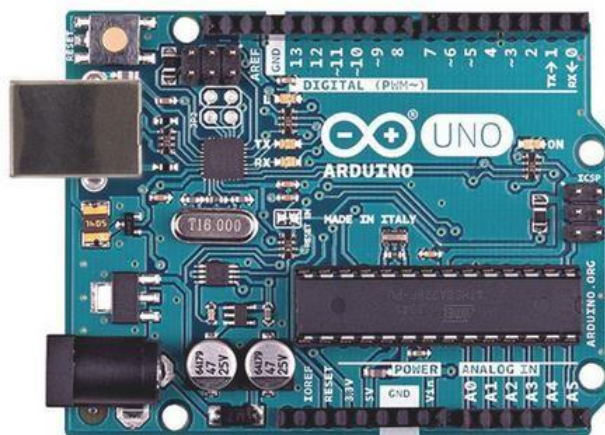


Figura 1 – Arduino UNO R3

Objetivo

O trabalho tem de dar resposta a dois objetivos, que devem constar no relatório:

1) Fazer o levantamento das características técnicas mais importantes como, por exemplo, os limites de corrente de input/output dos pinos do Arduino e alimentação apropriada.

2) Desenvolvimento do programa onde são aplicados os conhecimentos já adquiridos relativamente à linguagem de programação do Arduino (C/C++), de acordo com os requisitos listados no guia prático.

Aparato Experimental

- Arduino UNO R3;
- Sensor temperatura;
- Led's;
- Breadboard1;
- Resistores;
- Wires;

Sensor de temperatura

Mesmo se tratando de uma variação automática onde os valores e formulas foram inseridas a simular um valor e medições aleatórios, dentro da elaboração do circuito optamos por colocar o sensor para realização dos testes em modo de valores e medição reais se fosse o caso.

Usamos o sensor LM35 que é um sensor de precisão que apresenta uma saída de tensão linear proporcional à temperatura em que ele se encontrar no momento, tendo em sua saída um sinal de 10mV para cada grau Célsius de temperatura. Verificamos que o sensor não necessita de qualquer calibração externa para fornecer com exatidão os valores de temperatura com variações de $\frac{1}{4}^{\circ}\text{C}$ ou até mesmo $\frac{3}{4}^{\circ}\text{C}$ dentro da faixa de temperatura entre -55°C e 150°C , que está dentro da faixa de medição proposta inicialmente no projeto que é entre -40°C e 120°C .

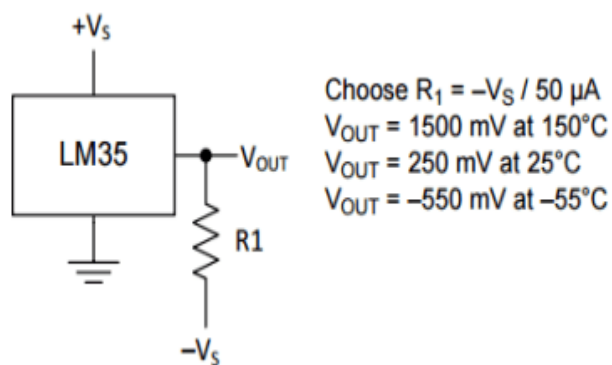


Figura 2 – LM35 – modo escala

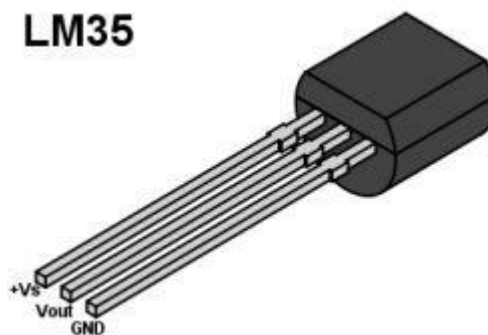


Figura 3 – LM35

Breadboard

Podemos traduzir por “tábua de pão”, esta “tábua” é usada como uma matriz de contatos que permite a elaboração e construção de circuitos para testes experimentais sem precisar de solda neste momento, permitindo fazer mudanças e organizações com rapidez e segurança partindo de uma alteração de posição de um determinado componente ou substituição, ainda podendo acrescentar ou retirar componentes para melhor funcionamento e obter resultados conforme o esperado.

A breadboard nos permite conectar uma série de dispositivos e componentes assim como: Circuitos integrados (CIs); Capacitores; Diodos; Led; Sensores; Resistores; Transistores.

Levando em consideração que seu funcionamento é por linha e colunas, observando as laterais onde são linhas positivas e negativas para transitar esses valores pela placa afim de facilitar e organizar os circuitos elaborados, assim como a montagem na placa dos circuitos devem seguir no meio a orientação das colunas e linhas para o funcionamento e conexão da corrente e do circuito.

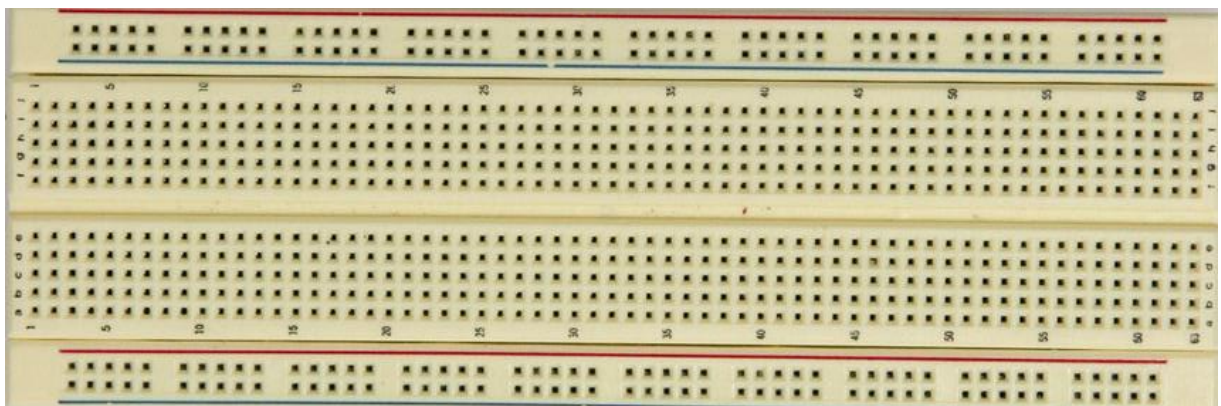


Figura 1.1 – Breadboard

Led

Podemos também chamar de diodo emissor de luz – led, dentro das pesquisas realizadas podemos identificar que ele é o mais simples componente eletrônico semicondutor, é composto de cristal de silício ou germânio. Existem vários tipos de diodos, para diversos tipos de aplicações, porém vamos nós reter ao LED, que é o que vamos usar no trabalho.

Possui a tecnologia utilizada nos chips dos computadores, tem a propriedade de transformar energia elétrica em luz. Tal transformação é diferente da encontrada nas lâmpadas convencionais que utilizam filamentos metálicos, radiação ultravioleta e descarga de gases. Os LED's, podem fazer a transformação de energia elétrica em luz que é feita na matéria, chamada de Estado Sólido.

O LED é um componente do tipo bipolar, tem um terminal chamado anodo e outro chamado catodo. Dependendo de como for polarizado, permite ou não a passagem de corrente elétrica e, conseqüentemente, a geração ou não de luz.

Vamos conseguir ver na imagem abaixo o componente mais importante de um LED é o chip semicondutor responsável pela geração de luz e seu controle, este chip tem dimensões muito reduzidas.

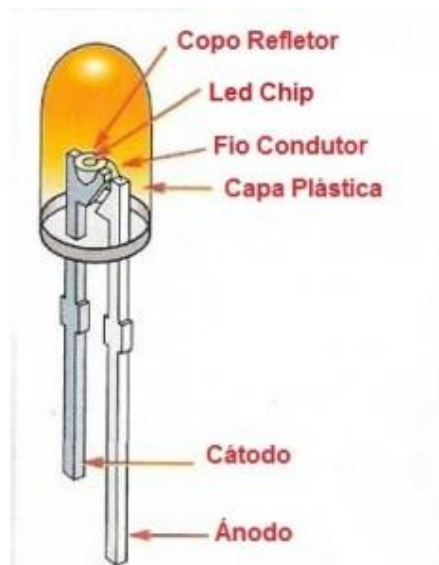


Figura 1.2 – Led

Resistores

Baseando em informações de pesquisa sobre Resistores podemos encontrar inúmeros tipos e modelos, assim como também há uma grande diversidade para suas atualizações, eles são componentes eletrônicos que tem como principal característica sua resistência elétrica, cuja ação é por limite ao fluxo de elétrons.

$$R = \frac{V}{I}$$

Figura 1.3 – formula resistência

R = Resistência elétrica (medida em Ohm);

I = Corrente (medida em Amperes);

V = Diferença de potencial (medida em Volts);

A equação é de suma importância pois através dela muitas outras equações podem ser obtidas.

Citação robocore tecnologias

A unidade Omega é representada pela letra grega chamada de Ohm, os esquemas elétricos resistores podem ser representados de duas formas:

American Style (R1);

Internacional Style (R2);

Resistores through-hole e surface-mount são tipos de resistores classificados tendo como base a forma que o componente é construído. Os through-hole, possuem terminais do componente que atravessam a PCB, mais utilizados no desenvolvimento de protótipos onde seus terminais se encaixam de forma mais fácil podendo ser alterado de lugar ou até mesmo feito teste sem precisar estar tendo complicações para o seu manuseio. O resistor surface-mount, já são soldados diretamente na superfície da placa, fazendo dessa característica um não tão vantajoso item para um processo de teste e criação,

porém na conclusão como a redução do espaço ocupado e maior produtividade na montagem de circuitos. Podemos utilizá-lo a fim de concluirmos o circuito e deixá-lo de forma mais enxuta.

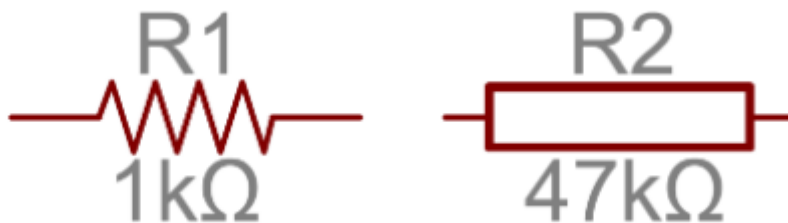


Figura 1.3 – resistors through-hole e surface mount

Seno

Ao elaborar o dispositivo conectamos a uma tensão de 5.0 V, o pino GND como negativo do Arduino para o circuito. O alinhamento dos pinos do dispositivo permite que usássemos os pinos GND, 2, como input do sensor, 4, 5, 6 e 7, para os led's do Arduino para o conectarmos diretamente esses periféricos no circuito.

“Variação sinusoidal entre valores arbitrários T_{min} e T_{max} , definidos como constantes, com ruído obtido a partir do gerador de números pseudo aleatórios, $random()$, de acordo com a seguinte expressão:

$$T = f(\sin(wt)) + A * \text{random}(R_{min}, R_{max})$$

em que $f()$ é a função que permite variar o sinal entre os limites T_{min} e T_{max} (uma transformação linear), $w = (2\pi)/P$, e a frequência angular (P o período), A a amplitude do ruído e R_{min} , R_{max} são os limites do gerador de números pseudo aleatórios (entre R_{min} e $R_{max}-1$, no caso da implementação da função $random()$ no Arduino). Estes parâmetros são constantes definidas pelo utilizador. “

Citação trabalho Física Aplicada, Mestre Nuno.

O parâmetro Value pode assumir valor HIGH, LOW, o que significa que se os valores enquadrarem dentro da regra definida o pino passa para estado

HIGH ou LOW, respectivamente onde foram selecionados 3 pinos, a saber digitalWrite (pino4); digitalWrite (pino5); digitalWrite (pino6), digitalWrite (pino7).

Dentro do parágrafo citado acima, onde podemos ver a função e os valor dado pelo professor para execução do trabalho temos a variável ruído onde temos o mínimo e máximo. Assim como teste de verificação apenas e isoladamente do ruído inserimos os valores de -3 e 3. Para verificação e obtenção de resultado e sua funcionalidade como poderá observar abaixo:

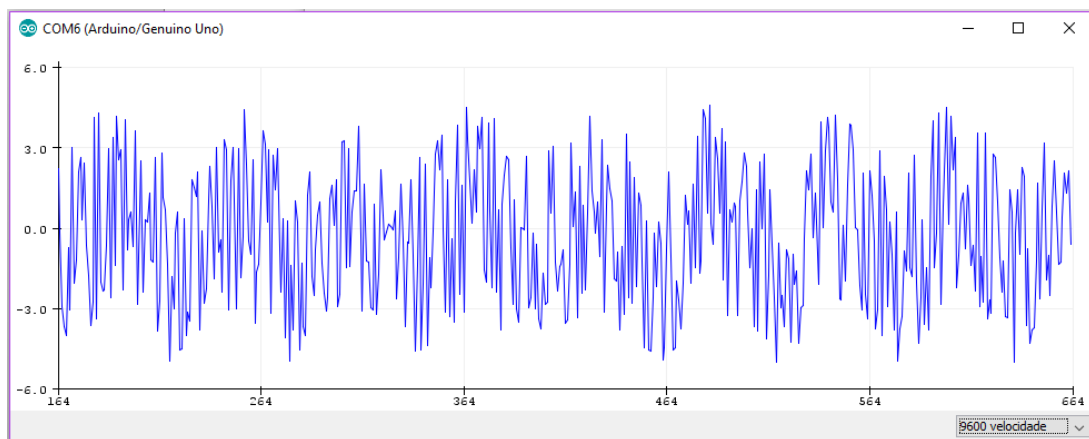


Figura 4 – ruído

```
serial_monitor_1 | Arduino 1.8.8 (Windows Store 1.8.19.0)
Arquivo Editar Sketch Ferramentas Ajuda

serial_monitor_1

if (t > 50 && t <= 120){
  digitalWrite(pino4, HIGH);
  digitalWrite(pino5, HIGH);
  digitalWrite(pino6, HIGH);
  Serial.print(F(" Led Violeta " ));
  Serial.print(t,2);
}

float ruido = 0.4 * random(-3,3);
t += 0.09;
y = calculation (t);
r = func(y) + ruido;

Serial.print(F(" t= ")); //imprima na tela a seguinte informação:
Serial.println(x);
Serial.print(F(" Tmin= " ));
Serial.print(-10);
Serial.print(F(" T0= " ));
Serial.print(10);
Serial.print(F(" T1= " ));
Serial.print(80);
Serial.print(F(" Tmax= " ));
Serial.print(110);
} //fim da void loop
```

Figura 5 – código para cálculo do ruído

```

serial_monitor_atual
133 if (temp >= 50 && temp <= 120){
134     digitalWrite(pino4, HIGH);
135     digitalWrite(pino5, HIGH);
136     digitalWrite(pino6, HIGH);
137     Serial.print(F("\t Led Violeta"));
138 }
139 if (temp > 120){
140     digitalWrite(pino7, HIGH); // para os valores de if luz de led acessa, estar
141     Serial.print(F("\t Led Marrom"));
142 }
143 #endif mserial
144 } //fim da void loop;
145 float func (float y){ // dividimos a equação de calculo e a variavel func es
146     f = (tmax - tmin) / 2.0 * y + tmax - (tmax - tmin) / 2.0;
147 return f;
148 }
149 float calculation(float temp){ // recebe o y e executa o calculo do seno;
150     y = sin(w*t);
151 return y;
152 }
153 #endif parteum
154
155 //-----

```

Figura 6 – código para cálculo do seno

Continuando a demonstração citada acima, onde podemos verificar a função e os valores dado pelo professor para execução do trabalho, temos a função onde vemos o mínimo e máximo dos valores demonstrado em sala de aula, laboratório prático. Assim também seguindo a aplicação do ruído, como teste de verificação apenas e isoladamente destes valores, verificamos e obtenção de resultado e sua funcionalidade como poderá observar abaixo:

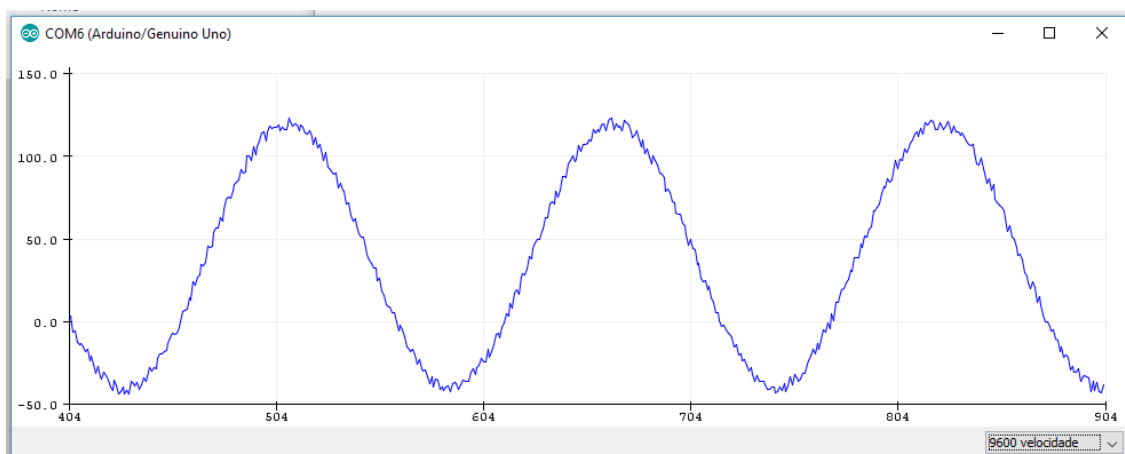


Figura 7 – seno temperatura

Na citação do manual do professor Nuno verificamos a solicitação para demonstrar através do MONITOR SERIAL os resultados obtidos.

Valores em demonstração: soma T em 0.09, Tmin, Tmax, T0, T1, cor LED, função t, conforme solicitação em manual de trabalho.

Para a segunda parte podemos alterar o valor T de 0.09 para MILLIS como orientação dada pelo professor Nuno em sala de aula, assim poderemos aplicar o conhecimento em vários fatores de simulação da função.

A seguir na figura 7 abaixo utilizamos no monitor serial a impressão para os devidos resultados com a sequência proposta em cores o led, tmax, tmin, t, que é o resultado da temperatura calculada e T que é o valor somatório de 0.09 e na figura 7.1 o T é millis conforme mudança orientada pelo professor Nuno.

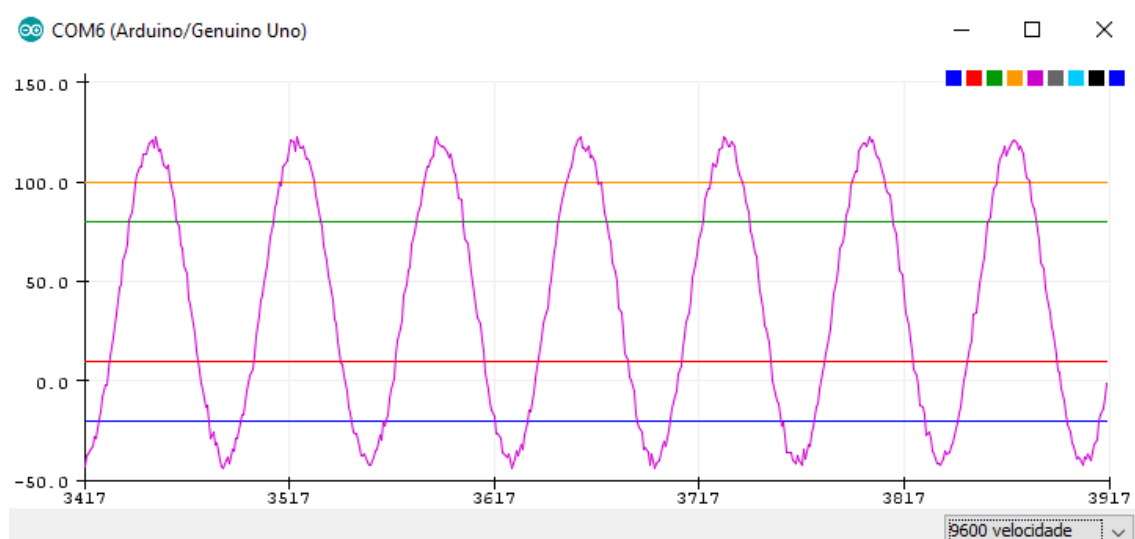


Figura 7.1 – seno temperatura com ruído aumentado e marcações de T0, T1, Tmax e Tmin.

Quando quisermos efetuar os testes para os valores em apresentação do monitor serial, teremos os seguintes resultados abaixo, como poderemos ver nas imagens para o valor t, foram utilizados dois tipos de exemplo:

Acréscimo com o 0.09;

Acréscimo em millis;

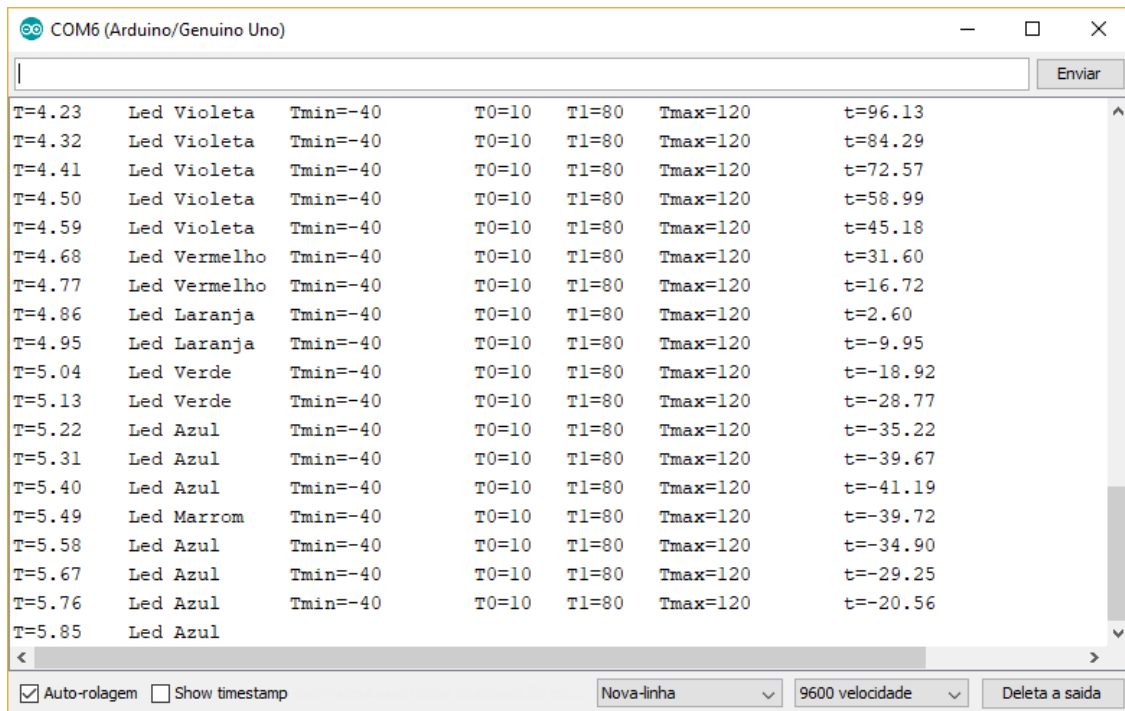


Figura 8 – Monitor Serial – $t = 0.09$

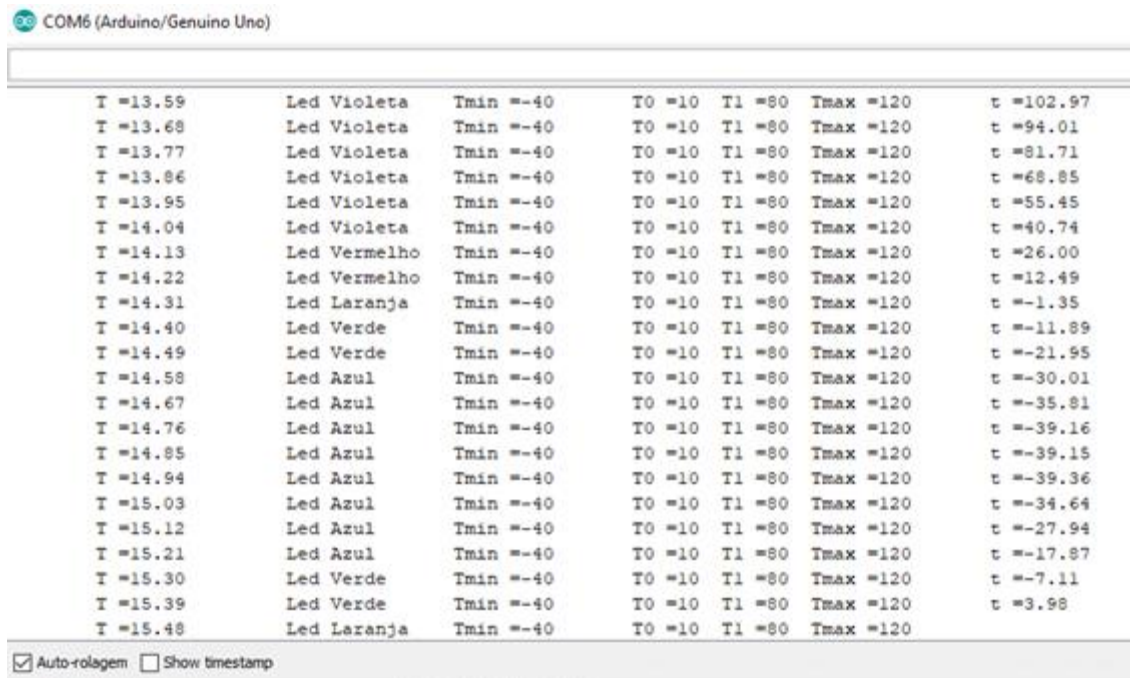


Figura 8.1 – Monitor Serial – $t = \text{millis}$

Código parte um

Esse é o código utilizado juntamente com a função para elaboração da primeira parte do projeto. A saber RUIDO e SENO.

```
#ifdef partenum

/* variáveis para cálculo e elaboração da função de leitura da temperatura, assim como calculo
* e regras a serem seguidas na estrutura de ação do código;
*/

float ruido = 0.8 * random(-3,3); // cálculo para o ruido;

t += 0.09; // acrescenta sempre o valor somado de + 0.09;

y = calculation (temp); //retorno do cálculo da função  $f = ((t_{max} - t_{min}) / 2.0) * y + (t_{max} - (t_{max} - t_{min}) / 2.0)$ ;

temp = func(y) + ruido; // calculo completo da função seno com ruido;

#ifdef mplotter

    Serial.print(F(" Tmax = " ));

    Serial.print(120); // marcação da linha de maior temperatura;

    Serial.print(F(" Tmin = " ));

    Serial.print(-40); // marcação da linha de menor temperatura;

    Serial.print(F(" t = "));

    Serial.println(temp); //imprima na tela o resultado da função completa;

    Serial.print(F(" T0 = " ));

    Serial.print(10); // marcação da linha de valor aleatório da temperatura;

    Serial.print(F(" T1 = " ));

    Serial.print(80); // marcação da linha de valor aleatório da temperatura;

#endif mplotter

#ifdef mserial

    Serial.print(F("\tTmin="));

    Serial.print(-40); // marcação da linha de menor temperatura;

    Serial.print(F("\tT0="));

    Serial.print(10); // marcação da linha de valor aleatório da temperatura;

    Serial.print(F("\tT1="));

    Serial.print(80); // marcação da linha de valor aleatório da temperatura;

    Serial.print(F("\tTmax="));

    Serial.print(120); // marcação da linha de maior temperatura;
```

```

    Serial.print(F("\ttt ="));

    Serial.println(temp); //imprima na tela o resultado da função completa;

    Serial.print(F("\tT ="));

    Serial.print(t,2); //imprimi o resultado t com acréscimo 0.09;

//Condição para resultados de temperaturas, com seleção dos pinos escolhidos e cores para representar a
temperatura;

if (temp < -40){ digitalWrite(pino7, HIGH); // para os valores de if luz de led acesa, estará HIGH;

    Serial.print(F("\t Led Marrom"));

}

if (temp >= -40 && temp < -20){

    digitalWrite(pino4, HIGH); // para os valores de if luz de led acesa, estará HIGH;

    digitalWrite(pino5, LOW); // para os valores de if luz de led apagada, estará LOW;

    digitalWrite(pino6, LOW);

    Serial.print(F("\t Led Azul"));

}

if (temp >= -20 && temp < 0){

    digitalWrite(pino4, LOW);

    digitalWrite(pino5, HIGH);

    digitalWrite(pino6, LOW);

    Serial.print(F("\t Led Verde"));

}

if (temp >= 0 && temp < 25){

    digitalWrite(pino4, LOW);

    digitalWrite(pino5, LOW);

    digitalWrite(pino6, HIGH);

    Serial.print(F("\t Led Laranja"));

}

if (temp >= 25 && temp < 50){

    digitalWrite(pino4, HIGH);

    digitalWrite(pino5, HIGH);

    digitalWrite(pino6, LOW);

    Serial.print(F("\t Led Vermelho"));

}

if (temp >= 50 && temp <= 120){

    digitalWrite(pino4, HIGH);

    digitalWrite(pino5, HIGH);

```



```

    digitalWrite(pino6, HIGH);

    Serial.print(F("\t Led Violeta"));
}

if (temp > 120){    digitalWrite(pino7, HIGH); // para os valores de if luz de led acesa, estará HIGH;

    Serial.print(F("\t Led Marrom"));
}

#endif mserial

#ifdef gnuplot

    Serial.println("#Trabalho Física Aplicada a Computação - Mestre Nuno\n");

    Serial.println("#Gerador de Seno - 1º Ano - FAC 2018-19\n");

    Serial.println("#tTmin\ttTmax\ttt\ttT0\ttT1\ttT");

    Serial.print(F("\tTmin ="));

    Serial.print(-40); // marcação da linha de menor temperatura;

    Serial.print(F("\tT0 ="));

    Serial.print(10); // marcação da linha de valor aleatório da temperatura;

    Serial.print(F("\tT1 ="));

    Serial.print(80); // marcação da linha de valor aleatório da temperatura;

    Serial.print(F("\tTmax ="));

    Serial.print(120); // marcação da linha de maior temperatura;

    Serial.print(F("\tT ="));

    Serial.println(temp); //imprima na tela o resultado da função completa;

    Serial.print(F("\tT ="));

    Serial.print(t,2); //imprimi o resultado t com acréscimo 0.09;

#endif gnuplot

} //fim da void loop;

float func (float y){

    f = (tmax - tmin) / 2.0 * y + tmax - (tmax - tmin) / 2.0;

return f;    // dividimos a equação de cálculo e a variável func está recebendo y e a parte do cálculo em variável f;

}

float calculation(float temp){

    y = sin(w*t);

return y;    // recebe o y e executa o cálculo do seno;

}

#endif parteum    //-----

```

Seno de 7

Segunda parte do projeto, onde deveríamos criar uma função que nós desse o resultado de um somatório de 7 senos. Baseando na contagem e reiniciando sempre de 7 em 7 dentro do somatório.

Utilizamos a seguinte formulação e código abaixo:

```
214 Serial.print(F("\tTl ="));
215 Serial.print(80); // marcação da linha de valor aleatorio da temperatura;
216 Serial.print(F("\tTmax ="));
217 Serial.print(120); // marcação da linha de maior temperatura;
218 Serial.print(F("\tT ="));
219 Serial.println(temp); //imprima na tela o resultado da função completa;
220 Serial.print(F("\tT ="));
221 Serial.print(t,2); //imprimi o resultado t com acrescimo em millis;
222 #endif gnuplot
223 } //fim da void loop;
224 float func (float y){
225     f = ((tmax - tmin) / 2.0) * y + (tmax - (tmax - tmin) / 2.0);
226 return f; // dividimos a equação de calculo e a variavel func esta recebendo y e a parte do calculo em variavel f;
227 }
228 float calculation(float temp){
229     y = 0.;
230 for( int k = 1; k < 8; k++){ //regra para seno de 7 valores;
231     y += (sin(k*w*t));
232 }
233 return y; // recebe o y e executa o calculo do seno;
234 }
235 #endif partedeois
236
237 //-----
```

Figura 9 – código 7 seno

Nesta segunda parte do projeto, segue o resultado de um somatório de 7 (sete) senos. Utilizamos recursos conforme enunciado e adequamos o mesmo para função do programa para o sinal ser gerado.

Será gerado não por uma função seno mas por uma soma de N funções; a saber 7 (sete) de acordo com a seguinte expressão:

$$T=f(\sum k = 1N \sin(kwt)) + A * \text{random}(Rmin, Rmax)$$

O parâmetro N, 7 (sete), que define o número de termos do somatório, e definido pelo utilizador. A figura 9.1 mostra um exemplo do sinal gerado a partir de uma série de 7 senos.

Citação trabalho Física Aplicada, Mestre Nuno, com pequenas adequações.

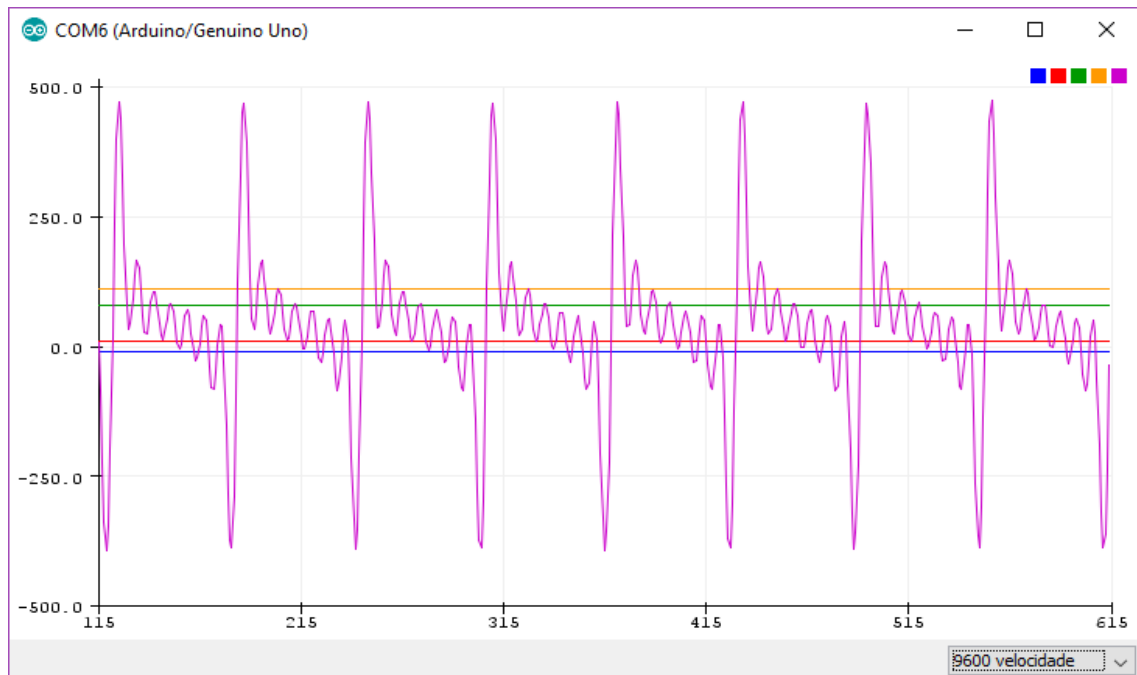


Figura 9.1 – seno de 7

Visando nesta segunda parte do projeto, com âmbito de conhecimento e aperfeiçoamento, inserimos dentro da função pedido a parte da impressão da media MCM para visualização da situação e curiosidade de funcionamento do projeto. Conforme imagem 9.2 abaixo.

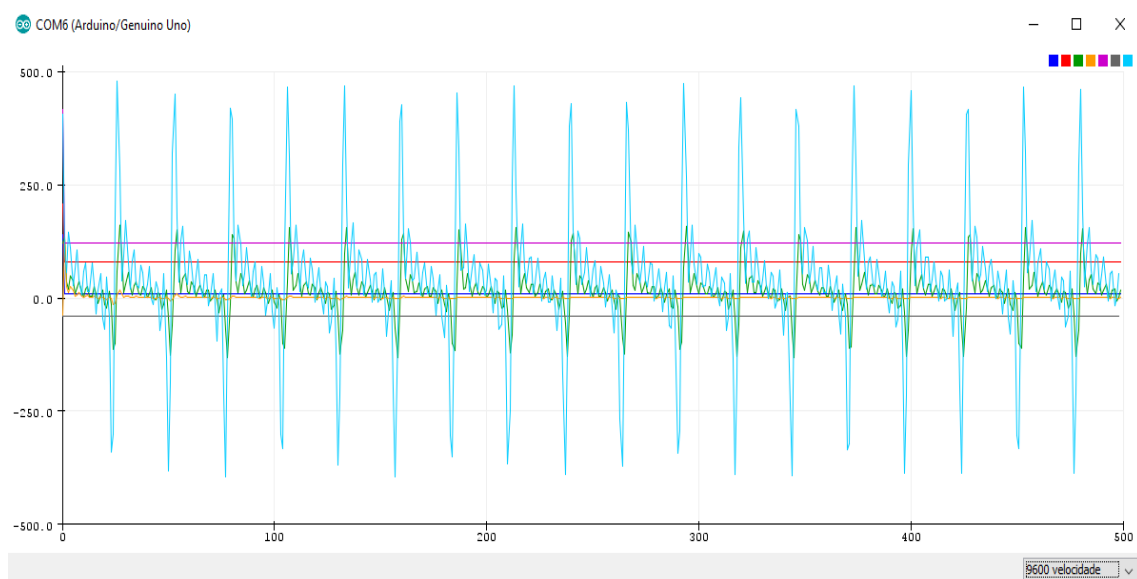


Figura 9.2 – Seno de 7 com media

Segue o código descrito para execução da segunda parte do projeto, conforme solicitação de valores 7 seno.

```
#ifdef partedois

    float ruido = 0.3 * random(-10,10); // cálculo para o ruido;

    t = (float) millis()/1000.; // acrescenta o cálculo/contagem em millis;

    y = calculation (temp); //retorno do cálculo da função  $f = ((t_{\text{max}} - t_{\text{min}}) / 2.0) * y + (t_{\text{max}} - (t_{\text{max}} - t_{\text{min}}) / 2.0)$ ;

    temp = func(y) + ruido; // calculo completo da função seno com ruido;

#ifdef mplotter

    Serial.print(F(" Tmax = " ));

    Serial.print(120); // marcação da linha de maior temperatura;

    Serial.print(F(" Tmin = " ));

    Serial.print(-40); // marcação da linha de menor temperatura;

    Serial.print(F(" t = "));

    Serial.println(temp); //imprima na tela o resultado da função completa;

    Serial.print(F(" T0 = " ));

    Serial.print(10); // marcação da linha de valor aleatório da temperatura;

    Serial.print(F(" T1 = " ));

    Serial.print(80); // marcação da linha de valor aleatório da temperatura;

#endif mplotter

#ifdef mserial

    Serial.print(F("\tt= ")); //imprima na tela o resultado da função completa;

    Serial.println(temp);

    Serial.print(F("\tTmin= "));

    Serial.print(-40); // marcação da linha de menor temperatura;

    Serial.print(F("\tT0= "));

    Serial.print(10); // marcação da linha de valor aleatório da temperatura;

    Serial.print(F("\tT1= "));

    Serial.print(80); // marcação da linha de valor aleatório da temperatura;
```

```

    Serial.print(F("\tTmax= "));

    Serial.print(120); // marcação da linha de maior temperatura;

    Serial.print(F("\tT= "));

    Serial.print(t,2); //imprimi o resultado t com millis;

#endif mserial

#ifdef gnuplot

    Serial.println("#Trabalho Física Aplicada a Computação - Mestre Nuno\n");

    Serial.println("#Gerador de Seno - 1º Ano - FAC 2018-19\n");

    Serial.println("#tTmin\tTmax\tt\tT0\tT1\tT");

    Serial.print(F("\tTmin ="));

    Serial.print(-40); // marcação da linha de menor temperatura;

    Serial.print(F("\tT0 ="));

    Serial.print(10); // marcação da linha de valor aleatório da temperatura;

    Serial.print(F("\tT1 ="));

    Serial.print(80); // marcação da linha de valor aleatório da temperatura;

    Serial.print(F("\tTmax ="));

    Serial.print(120); // marcação da linha de maior temperatura;

    Serial.print(F("\tt ="));

    Serial.println(temp); //imprima na tela o resultado da função completa;

    Serial.print(F("\tT ="));

    Serial.print(t,2); //imprimi o resultado t com acréscimo em millis;

#endif gnuplot

} //fim da void loop;

float func (float y){

    f = ((tmax - tmin) / 2.0) * y + (tmax - (tmax - tmin) / 2.0);

return f;    // dividimos a equação de cálculo e a variável func está recebendo y e a parte do
cálculo em variável f;

}

float calculation(float temp){

    y = 0.;

for( int k = 1; k < 8; k++){    //regra para seno de 7 valores;

```

```

        y += (sin(k*w*t));
    }
    return y;    // recebe o y e executa o cálculo do seno;
}

#endif partedois

//-----

```

Media final MCM

O objetivo das médias a seguir é ir somando todos os valores apresentados ou gerados em temperatura, após resultados das funções, valores inseridos ou determinados para o ruído por exemplo e depois dividir pelo número de contador de medições realizada.

Sendo assim podemos obter os seguintes resultados com a formula, código.

```

244 #ifdef mediac          //media acumulada, soma todos os valores apresentados em temp e depois divide pelo numero do contador;
245     temperaturas += temp;
246     media = temperaturas/contador;
247     contador++;
248 #endif mediac
249 #ifdef mmmedia          //media corrente, soma 3 valores apresentados em temp e depois divide por 3;
250     tempContLeitura[contador] = temp;
251     contador++;
252     if(contador == contLeitura) {
253         mmm = 0.;
254         for(int k = 0; k < contLeitura; k++){          //condição e contador para a apresentação da media em 3 em 3;
255             mmm = mmm + tempContLeitura[k];
256         }
257         mmm = mmm / contLeitura;
258         contador = 0;
259     }

```

Figura 10 – código mcm

Acima usamos a função para o cálculo da média corrente e da média das últimas M medições.

- Media (MCM): $T_{med} = (T1 + T2 + \dots + Tk) / k$, k e o número de medições realizadas;

A seguir utilizaremos a função para o cálculo da média acumulativa e da média das últimas M medições.

- Media (MMM): $TM = (Tk + Tk+1 + \dots + Tk+M-1) / M$, M e número medições que entram no cálculo.

Citação trabalho Física Aplicada, Mestre Nuno.

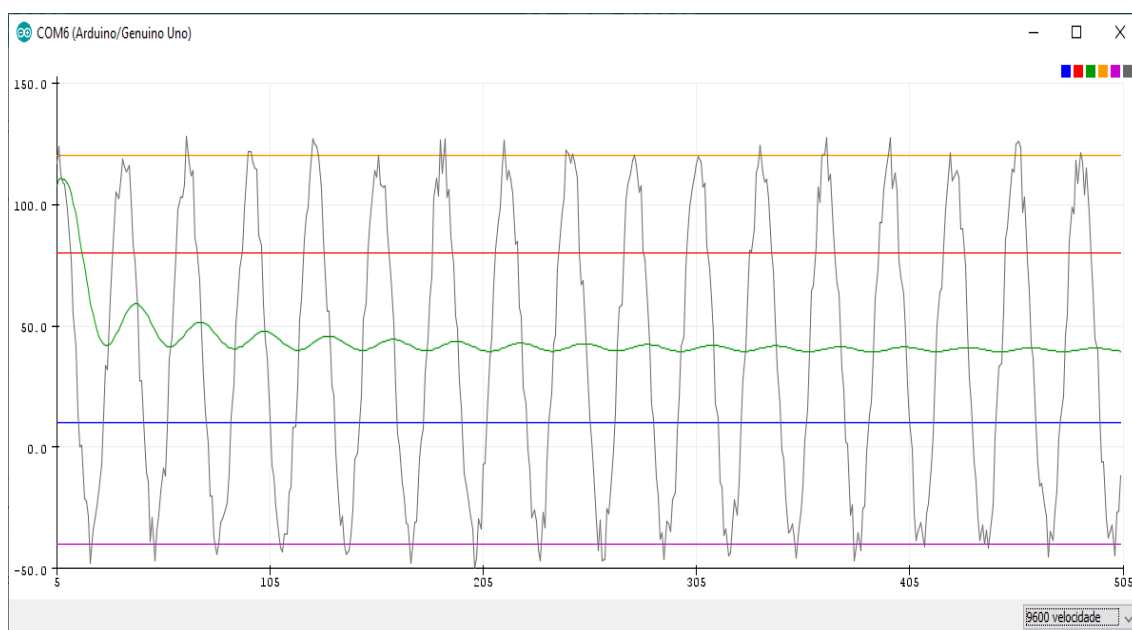


Figura 10.1 – media acumulativa mcm

COM6 (Arduino/Genuino Uno)

Enviar

Tmin= -40	Tmax= 120	T= 12.49	Contador= 95	Media= 38.79	Led Laranja	T0= 0.12	T1= 80	t= 48.20
Tmin= -40	Tmax= 120	T= 12.62	Contador= 96	Media= 38.89	Led Vermelho	T0= 0.38	T1= 80	t= 74.06
Tmin= -40	Tmax= 120	T= 12.76	Contador= 97	Media= 39.26	Led Violeta	T0= 0.60	T1= 80	t= 97.37
Tmin= -40	Tmax= 120	T= 12.89	Contador= 98	Media= 39.86	Led Violeta	T0= 0.79	T1= 80	t= 96.42
Tmin= -40	Tmax= 120	T= 13.02	Contador= 99	Media= 40.43	Led Violeta	T0= 0.92	T1= 80	t= 118.95
Tmin= -40	Tmax= 120	T= 13.16	Contador= 100	Media= 41.23	Led Violeta	T0= 0.99	T1= 80	t= 128.41
Tmin= -40	Tmax= 120	T= 13.29	Contador= 101	Media= 42.10	Led Marrom	T0= 0.99	T1= 80	t= 119.17
Tmin= -40	Tmax= 120	T= 13.42	Contador= 102	Media= 42.86	Led Violeta	T0= 0.92	T1= 80	t= 105.23
Tmin= -40	Tmax= 120	T= 13.56	Contador= 103	Media= 43.47	Led Violeta	T0= 0.78	T1= 80	t= 99.08
Tmin= -40	Tmax= 120	T= 13.69	Contador= 104	Media= 44.01	Led Violeta	T0= 0.58	T1= 80	t= 90.37
Tmin= -40	Tmax= 120	T= 13.83	Contador= 105	Media= 44.46	Led Violeta	T0= 0.35	T1= 80	t= 64.61
Tmin= -40	Tmax= 120	T= 13.96	Contador= 106	Media= 44.65	Led Violeta	T0= 0.08	T1= 80	t= 53.74
Tmin= -40	Tmax= 120	T= 14.10	Contador= 107	Media= 44.74	Led Violeta	T0= -0.18	T1= 80	t= 30.55
Tmin= -40	Tmax= 120	T= 14.23	Contador= 108	Media= 44.60	Led Vermelho	T0= -0.44	T1= 80	t= 8.94
Tmin= -40	Tmax= 120	T= 14.36	Contador= 109	Media= 44.27	Led Laranja	T0= -0.66	T1= 80	t= -17.73
Tmin= -40	Tmax= 120	T= 14.50	Contador= 110	Media= 43.70	Led Verde	T0= -0.83	T1= 80	t= -32.78
Tmin= -40	Tmax= 120	T= 14.63	Contador= 111	Media= 43.01	Led Azul	T0= -0.95	T1= 80	t= -32.01
Tmin= -40	Tmax= 120	T= 14.76	Contador= 112	Media= 42.33	Led Azul	T0= -1.00	T1= 80	t= -48.90
Tmin= -40	Tmax= 120	T= 14.90	Contador= 113	Media= 41.52	Led Marrom	T0= -0.98	T1= 80	t= -47.13
Tmin= -40	Tmax= 120	T= 15.03	Contador= 114	Media= 40.73	Led Marrom	T0= -0.88	T1= 80	t= -34.70
Tmin= -40	Tmax= 120	T= 15.17	Contador= 115	Media= 40.07	Led Azul	T0= -0.73	T1= 80	t= -13.49
Tmin= -40	Tmax= 120	T= 15.30	Contador= 116	Media= 39.61	Led Verde	T0= -0.52	T1= 80	t= -11.95

☐ Auto-rolagem☐ Show timestamp

Nova-linha

9600 velocidade

Deleta a saída

Figura 10.2 – media mcm – monitor serial

Media final MMM

Nesta parte do projeto teremos que calcular a mmm que é a média somando 3 em 3 temperaturas e dando o resultado subsequente em seguida e assim correndo de forma continua enquanto a temperatura está sendo medida.

Com isso obtemos os seguintes resultados abaixo demonstrados, através do código em seguida e da formula citada acima em quadro anterior.

```
239 #ifndef partetres
240     float ruido = 1 * random(-10,10); // calculo para o ruido;
241     t = (float) millis()/1000.0; // acrescenta o calculo/contagem em millis;
242     y = calculation (temp); //retorno do calculo da função f = ((tmax - tmin) / 2.0) * y + (tmax - (tmax - tmin) / 2.0);
243     temp = func(y) + ruido; // calculo completo da função seno com ruido;
244 #endif mediad
245     temperaturas += temp;
246     media = temperaturas/contador;
247     contador++;
248 #endif mediad
249 #ifndef mmmmedia //media corrente, soma 3 valores apresentados em temp e depois divide por 3;
250     tempContLeitura[contador] = temp;
251     contador++;
252     if(contador == contLeitura) {
253         mmm = 0.;
254         for(int k = 0; k < contLeitura; k++){ //condição e contador para a apresentação da media em 3 em 3;
255             mmm = mmm + tempContLeitura[k];
256         }
257         mmm = mmm / contLeitura;
258         contador = 0;
259     }
260 #endif mmmmedia
261 #endif suav
```

Figura 11 – código calculo media mmm

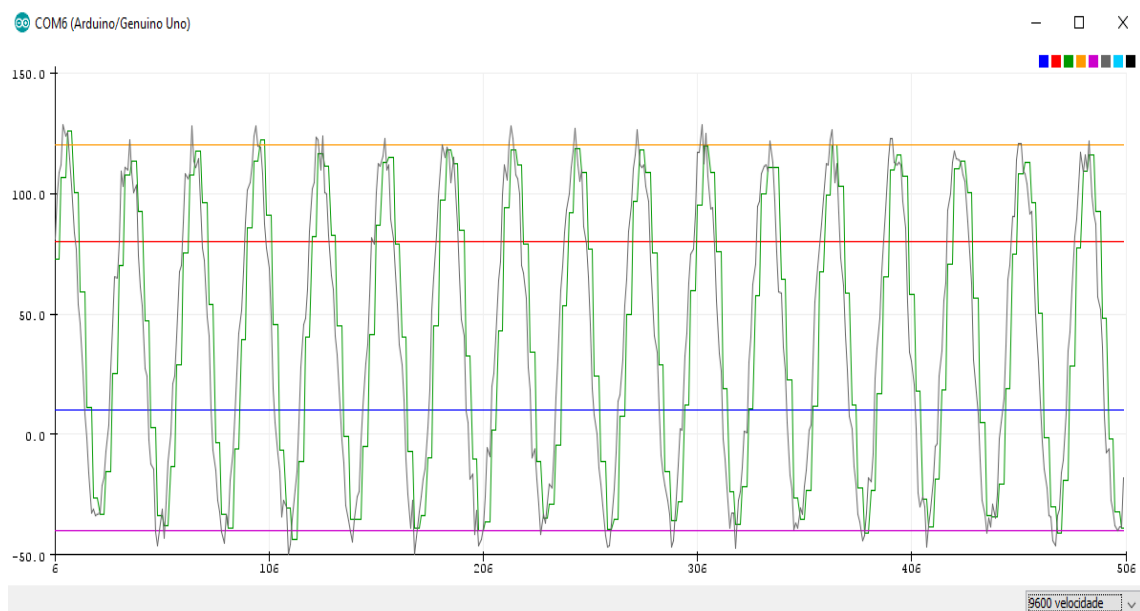


Figura 11.1 – media mmm – plotter

COM6 (Arduino/Genuino Uno)

Enviar

Tmin= -40	Tmax= 120	T= 14.14	MMM= 74.82	Led Vermelho	T0= -0.23	T1= 80	t= 29.48
Tmin= -40	Tmax= 120	T= 14.25	MMM= 74.82	Led Vermelho	T0= -0.45	T1= 80	t= 6.79
Tmin= -40	Tmax= 120	T= 14.37	MMM= 23.15	Led Laranja	T0= -0.65	T1= 80	t= -16.76
Tmin= -40	Tmax= 120	T= 14.49	MMM= 23.15	Led Verde	T0= -0.81	T1= 80	t= -27.49
Tmin= -40	Tmax= 120	T= 14.61	MMM= 23.15	Led Azul	T0= -0.92	T1= 80	t= -39.71
Tmin= -40	Tmax= 120	T= 14.72	MMM= -27.99	Led Azul	T0= -0.99	T1= 80	t= -34.89
Tmin= -40	Tmax= 120	T= 14.84	MMM= -27.99	Led Azul	T0= -1.00	T1= 80	t= -30.82
Tmin= -40	Tmax= 120	T= 14.96	MMM= -27.99	Led Azul	T0= -0.96	T1= 80	t= -27.41
Tmin= -40	Tmax= 120	T= 15.07	MMM= -31.04	Led Azul	T0= -0.86	T1= 80	t= -35.75
Tmin= -40	Tmax= 120	T= 15.19	MMM= -31.04	Led Azul	T0= -0.72	T1= 80	t= -17.50
Tmin= -40	Tmax= 120	T= 15.31	MMM= -31.04	Led Verde	T0= -0.54	T1= 80	t= 1.10
Tmin= -40	Tmax= 120	T= 15.43	MMM= -17.38	Led Laranja	T0= -0.32	T1= 80	t= 20.08
Tmin= -40	Tmax= 120	T= 15.54	MMM= -17.38	Led Laranja	T0= -0.09	T1= 80	t= 36.82
Tmin= -40	Tmax= 120	T= 15.66	MMM= -17.38	Led Vermelho	T0= 0.15	T1= 80	t= 46.80
Tmin= -40	Tmax= 120	T= 15.78	MMM= 34.57	Led Vermelho	T0= 0.37	T1= 80	t= 64.98
Tmin= -40	Tmax= 120	T= 15.90	MMM= 34.57	Led Violeta	T0= 0.58	T1= 80	t= 88.36
Tmin= -40	Tmax= 120	T= 16.02	MMM= 34.57	Led Violeta	T0= 0.75	T1= 80	t= 93.32
Tmin= -40	Tmax= 120	T= 16.13	MMM= 82.22	Led Violeta	T0= 0.88	T1= 80	t= 115.78
Tmin= -40	Tmax= 120	T= 16.25	MMM= 82.22	Led Violeta	T0= 0.97	T1= 80	t= 112.54
Tmin= -40	Tmax= 120	T= 16.37	MMM= 82.22	Led Violeta	T0= 1.00	T1= 80	t= 127.99
Tmin= -40	Tmax= 120	T= 16.49	MMM= 118.77	Led Marrom	T0= 0.98	T1= 80	t= 120.01
Tmin= -40	Tmax= 120	T= 16.60	MMM= 118.77	Led Marrom	T0= 0.90	T1= 80	t= 103.71

☐ Auto-rolagem

☐ Show timestamp

Nova-linha

9600 velocidade

Deleta a saída

Figura 11.2 – media mmm – monitor serial

Circuito e esquemático em fritzing

Concluindo a etapa do projeto, elaboramos o circuito em fritzing para visualização de como seria o circuito e o esquemático para essa solução.

Segue abaixo imagem elaborada dos mesmos.

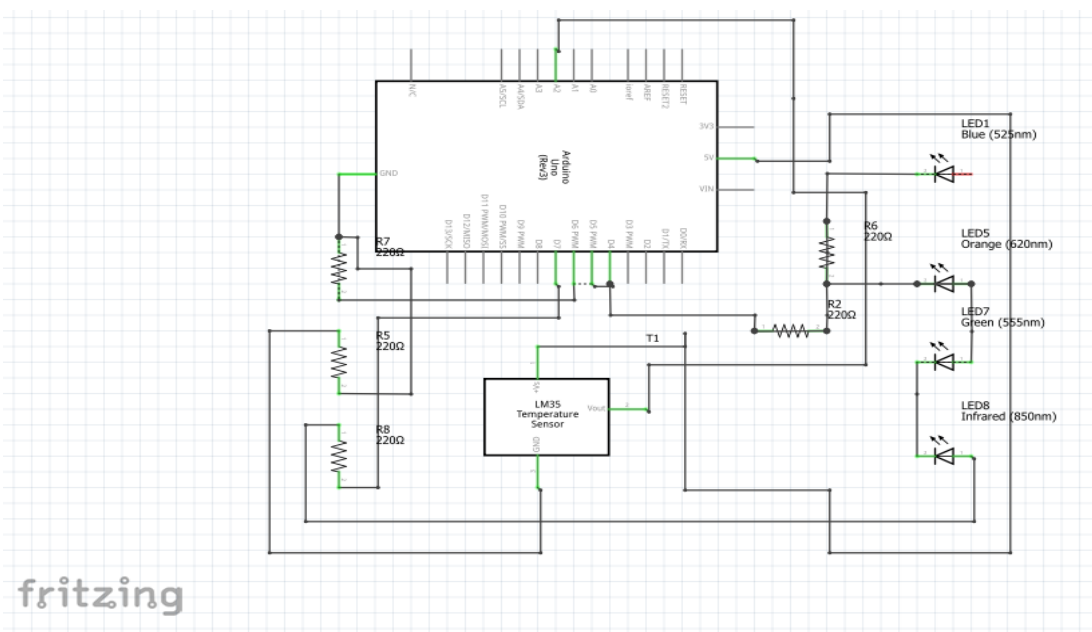


Figura 12 – modelo esquemático

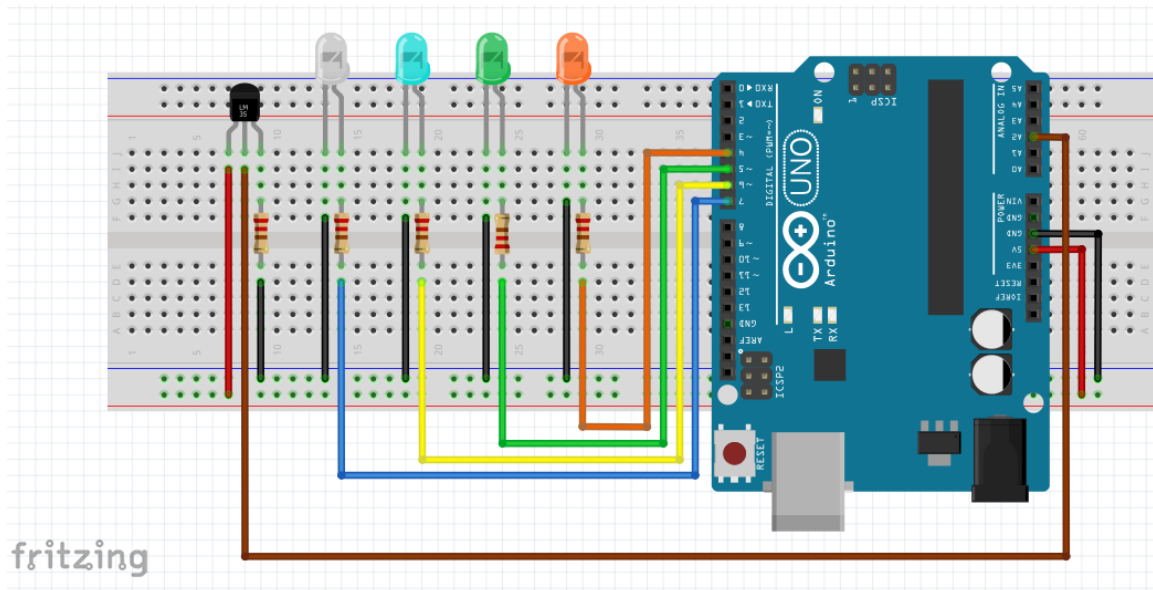


Figura 12.1 – modelo protoboard

Segue o código descrito para execução da terceira parte do projeto, conforme solicitação de valores de media MCM, MMM, SUAVIZAÇÃO.

```
#ifndef partetres

float ruido = 1 * random(-10,10); // cálculo para o ruido;

t = (float) millis()/1000.0; // acrescenta o cálculo/contagem em millis;

y = calculation (temp); //retorno do cálculo da função f = ((tmax - tmin) / 2.0) * y + (tmax -
(tmax - tmin) / 2.0);

temp = func(y) + ruido; // calculo completo da função seno com ruido;

#endif mediac //média acumulada, soma todos os valores apresentados em temp e
depois divide pelo número do contador;

temperaturas += temp;

media = temperaturas/contador;

contador++;

#endif mediac

//media corrente, soma 3 valores apresentados em temp e depois
divide por 3;

tempContLeitura[contador] = temp;

contador++;

if(contador == contLeitura) {
```

```

    mmm = 0.;
for(int k = 0; k < contLeitura; k++){    //condição e contador para a apresentação da média em
3 em 3;

    mmm = mmm + tempContLeitura[k];
}

    mmm = mmm / contLeitura;
    contador = 0;
}

#endif mmmedia

#ifdef suav
if (temp > 100){
    mms = temp * 0.5; //suavização parte superior;
}

if (temp < -20){
    mms = temp * 0.25; //suavização parte inferior;
}

#endif suav

#ifdef mplotter

    Serial.print(F(" T0 = " ));
    Serial.print(10); // marcação da linha de valor aleatório da temperatura;
    Serial.print(F(" T1 = " ));
    Serial.print(80); // marcação da linha de valor aleatório da temperatura;
    Serial.print(F(" Tmax = " ));
    Serial.print(120); // marcação da linha de maior temperatura;
    Serial.print(F(" Tmin = " ));
    Serial.print(-40); // marcação da linha de menor temperatura;
    Serial.print(F(" t = "));
    Serial.println(temp); //imprima na tela o resultado da função completa;
    Serial.print(F(" Media = "));
    Serial.print(media); // imprimir media mcm acumulativa;
    Serial.print(F("\tMMM= "));

```

```

    Serial.print(mmm); // imprime media mmm;
#endif mplotter

#ifdef mserial

    Serial.print(F("\tT0= "));

    Serial.print(y); // marcação da linha de valor aleatório da temperatura;

    Serial.print(F("\tT1= "));

    Serial.print(80); // marcação da linha de valor aleatório da temperatura;

    Serial.print(F("\tt= "));

    Serial.println(temp); //imprima na tela a função completa;

    Serial.print(F("\tTmin= "));

    Serial.print(-40); // marcação da linha de menor temperatura;

    Serial.print(F("\tTmax= "));

    Serial.print(120); // marcação da linha de maior temperatura;

    Serial.print(F("\tT= "));

    Serial.print(t,2); //imprimi o resultado T com millis;

    //Serial.print(F("\tContador= "));

    //Serial.print(contador); // imprimi a contagem;

    //Serial.print(F("\tMedia= "));

    //Serial.print(media); // imprimi media mcm acumulativa;

    Serial.print(F("\tMMM= "));

    Serial.print(mmm); // imprimi media mmm;

    //Serial.print(F("\tMMS= "));

    //Serial.print(mms); // imprimi suavização;

#endif gnuplot

    Serial.println("#Trabalho Física Aplicada a Computação - Mestre Nuno\n");

    Serial.println("#Gerador de Seno - 1º Ano - FAC 2018-19\n");

    Serial.println("#tTmin\tTmax\tt\tT0\tT1\tT");

    Serial.print(F("\tTmin ="));

    Serial.print(-40); // marcação da linha de menor temperatura;

    Serial.print(F("\tT0 ="));

    Serial.print(10); // marcação da linha de valor aleatório da temperatura;

```

```

Serial.print(F("\tT1 ="));

Serial.print(80); // marcação da linha de valor aleatório da temperatura;

Serial.print(F("\tTmax ="));

Serial.print(120); // marcação da linha de maior temperatura;

Serial.print(F("\tt ="));

Serial.println(temp); //imprima na tela o resultado da função completa;

Serial.print(F("\tT ="));

Serial.print(t,2); //imprimi o resultado t com acréscimo em millis;

Serial.print(F("\tMedia= "));

Serial.print(media);          // imprimi media mcm acumulativa;

Serial.print(F("\tMMM= "));

Serial.print(mmm); // imprimi media mmm;

#endif gnuplot

//Condição para resultados de temperaturas, com seleção dos pinos escolhidos e cores para
representar a temperatura;

if (temp < -40){    digitalWrite(pino7, HIGH); // para os valores de if luz de led acesa, estará
HIGH;

    Serial.print(F("\t Led Marrom"));

}

if (temp >= -40 && temp < -20){

    digitalWrite(pino4, HIGH); // para os valores de if luz de led acesa, estará HIGH;

    digitalWrite(pino5, LOW); // para os valores de if luz de led apagada, estará LOW;

    digitalWrite(pino6, LOW);

    Serial.print(F("\t Led Azul"));

}

if (temp >= -20 && temp < 0){

    digitalWrite(pino4, LOW);

    digitalWrite(pino5, HIGH);

    digitalWrite(pino6, LOW);

    Serial.print(F("\t Led Verde"));

}

if (temp >= 0 && temp < 25){

```

```

    digitalWrite(pino4, LOW);
    digitalWrite(pino5, LOW);
    digitalWrite(pino6, HIGH);
    Serial.print(F("\t Led Laranja"));
}

if (temp >= 25 && temp < 50){
    digitalWrite(pino4, HIGH);
    digitalWrite(pino5, HIGH);
    digitalWrite(pino6, LOW);
    Serial.print(F("\t Led Vermelho"));
}

if (temp >= 50 && temp <= 120){
    digitalWrite(pino4, HIGH);
    digitalWrite(pino5, HIGH);
    digitalWrite(pino6, HIGH);
    Serial.print(F("\t Led Violeta"));
}

if (temp > 120){    digitalWrite(pino7, HIGH); // para os valores de if luz de led acessa, estará
HIGH;

    Serial.print(F("\t Led Marrom"));
}

#endif mserial

} //fim da void loop;

float func (float y){    f = ((tmax - tmin) / 2.0) * y + (tmax - (tmax - tmin) / 2.0);

return f;                // dividimos a equação de cálculo e a variável func está recebendo y e a
parte do cálculo em variável f;

}    float calculation(float temp){    y = sin(w*t);

return y;                // recebe o y e executa o cálculo do seno;

}

#endif partetres

```

GNU PLOT

Imagens geradas dentro do gnuplot, para sequência dos valores obtidos durante o trabalho.

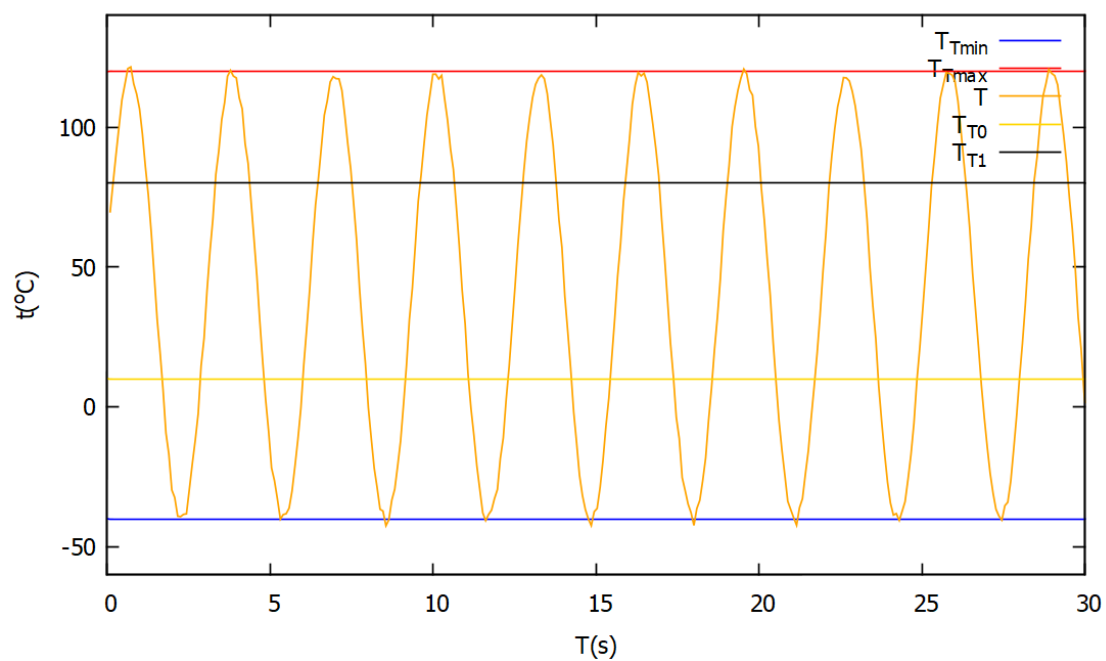


Figura – gnuplot seno

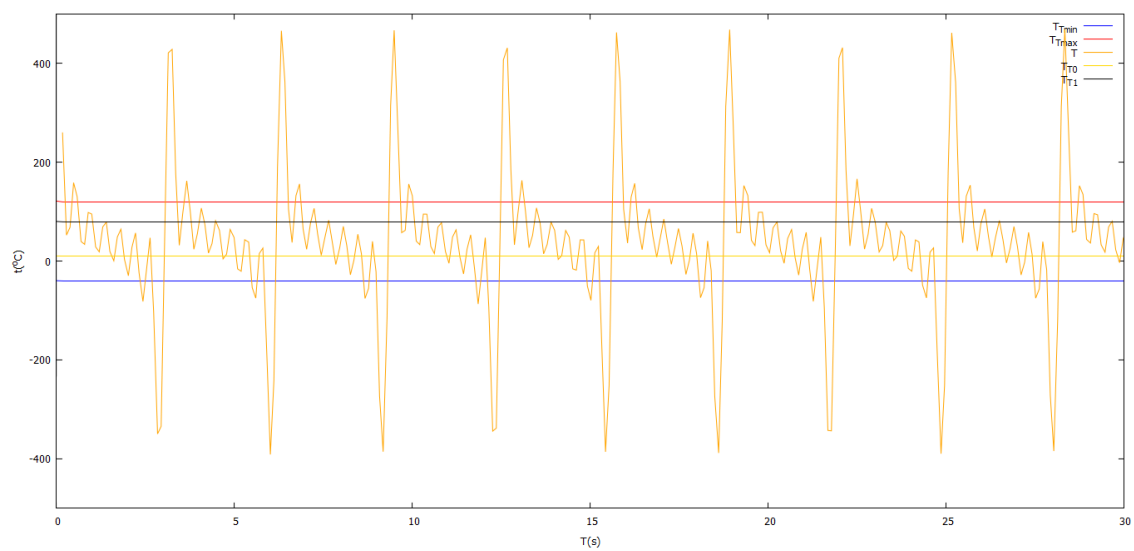


Figura – gnuplot seno 7

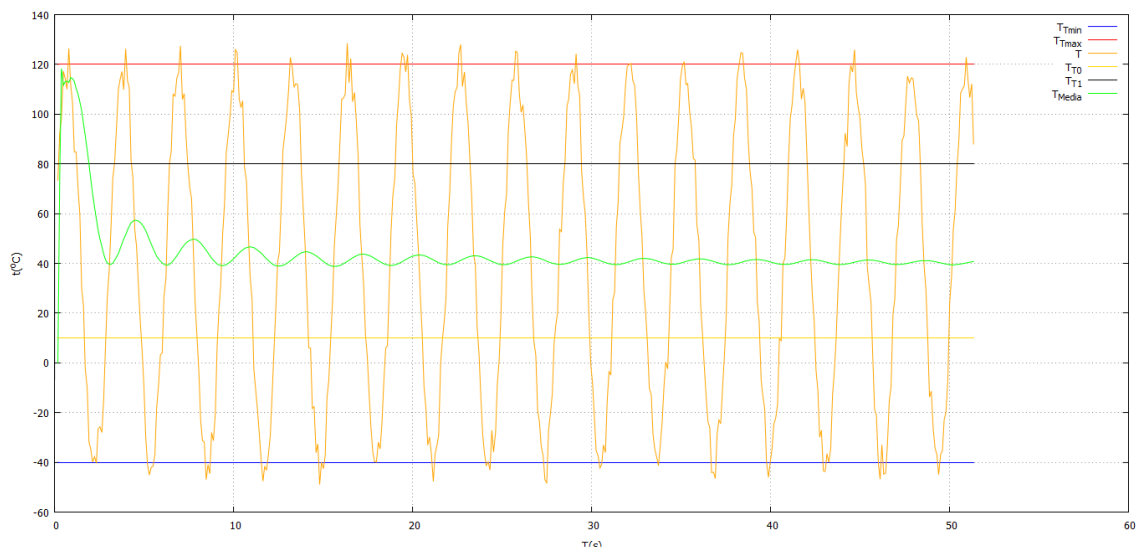


Figura – gnuplot seno media

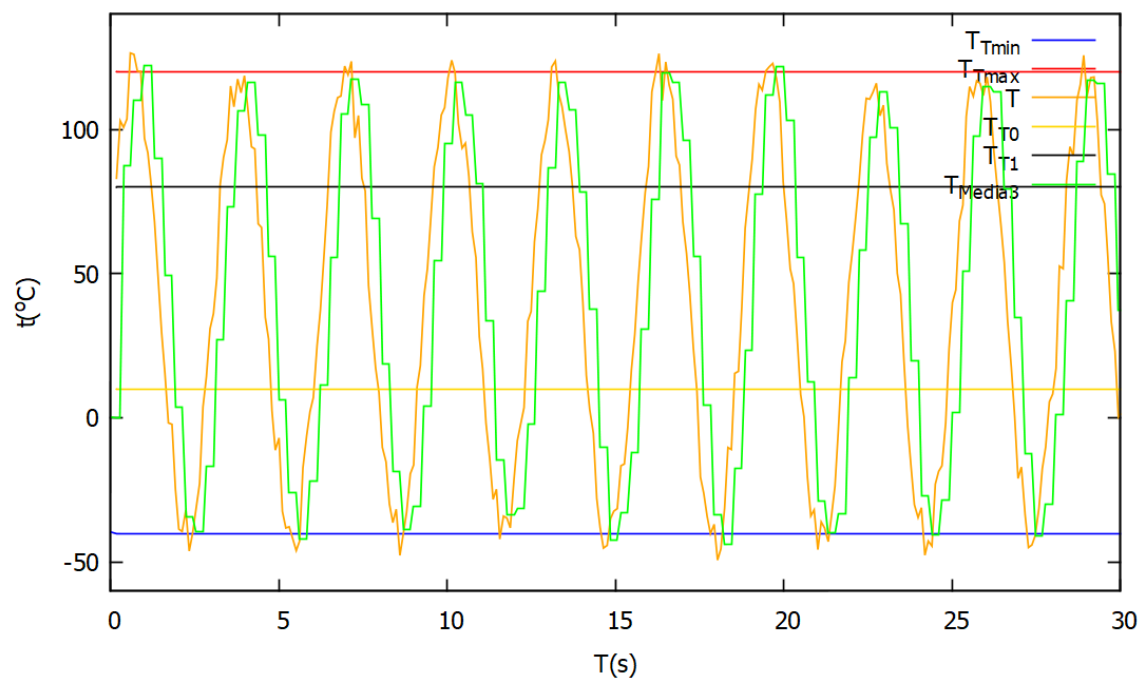


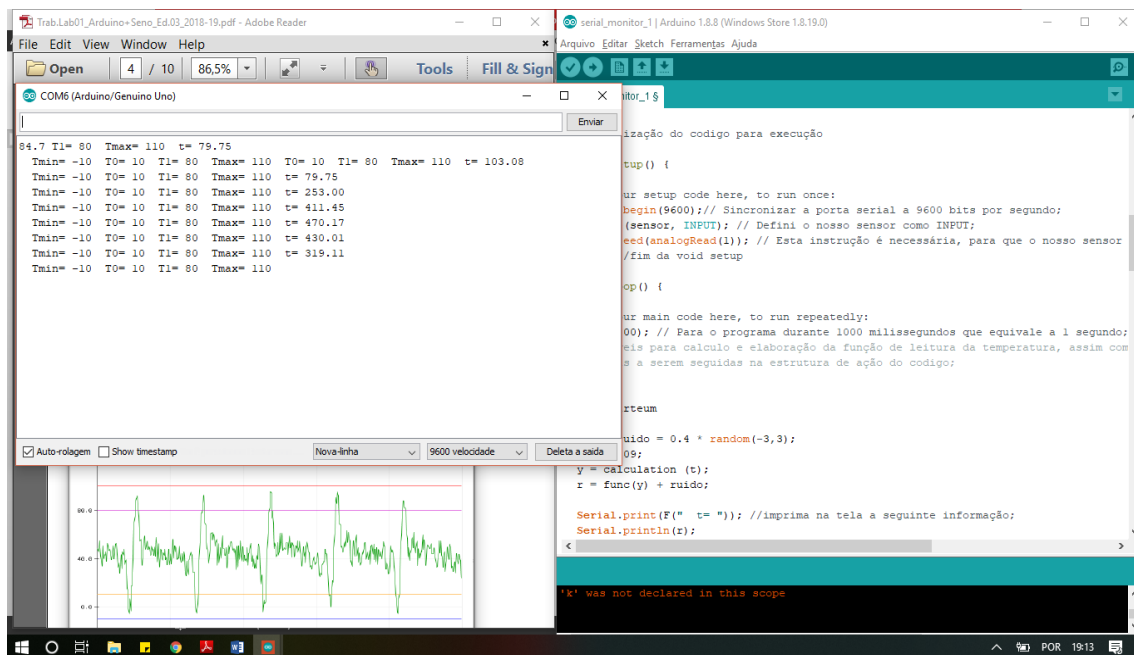
Figura – gnuplot seno media com 3

TESTES DE EXECUÇÃO

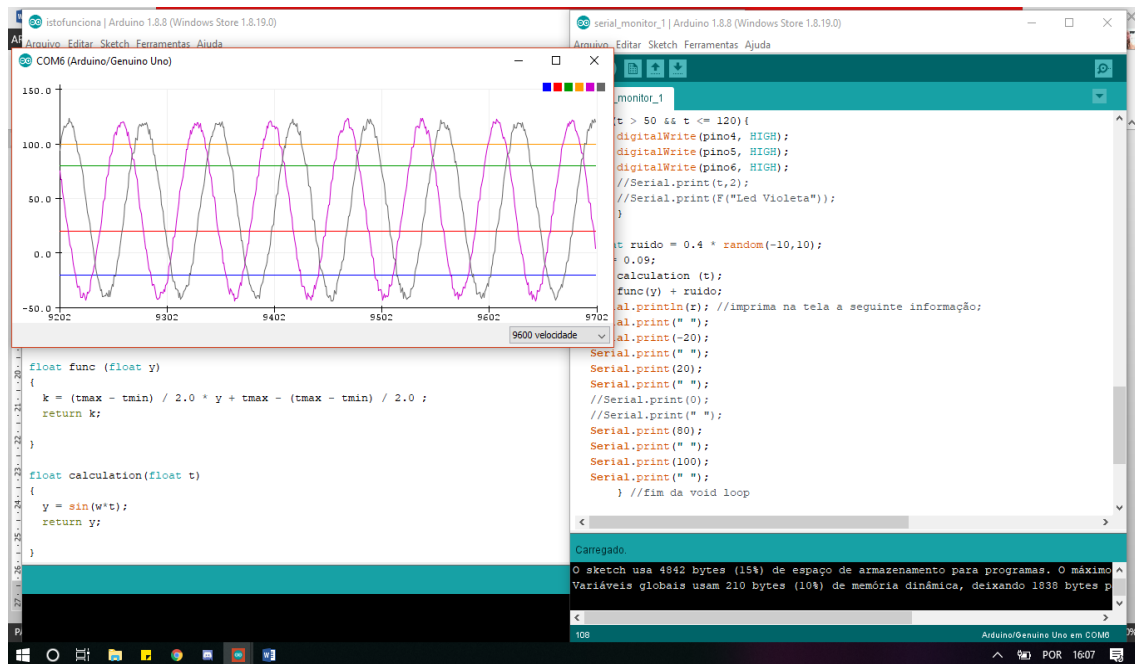
Durante o processo de elaboração e desenvolvimento tivemos vários testes de execução em todas as áreas do projeto e código onde aos poucos fomos corrigindo os erros encontrados e melhorando tanto o aspecto estrutural como a adaptação do funcionamento das funções, cálculos e variáveis para as devidas impressões e resultados em serial monitor, plotter e gnuplot. Sendo assim efetuamos os print's para visualização e guardar o histórico do processo de desenvolvimento.

Demonstramos a seguir por algumas telas/print o desenvolvimento e testes/erros no decorrer do processo de elaboração do código.

Print teste somatório de temperatura – monitor serial



Print teste ruído com temperatura – monitor serial



Print teste somatório – plotter

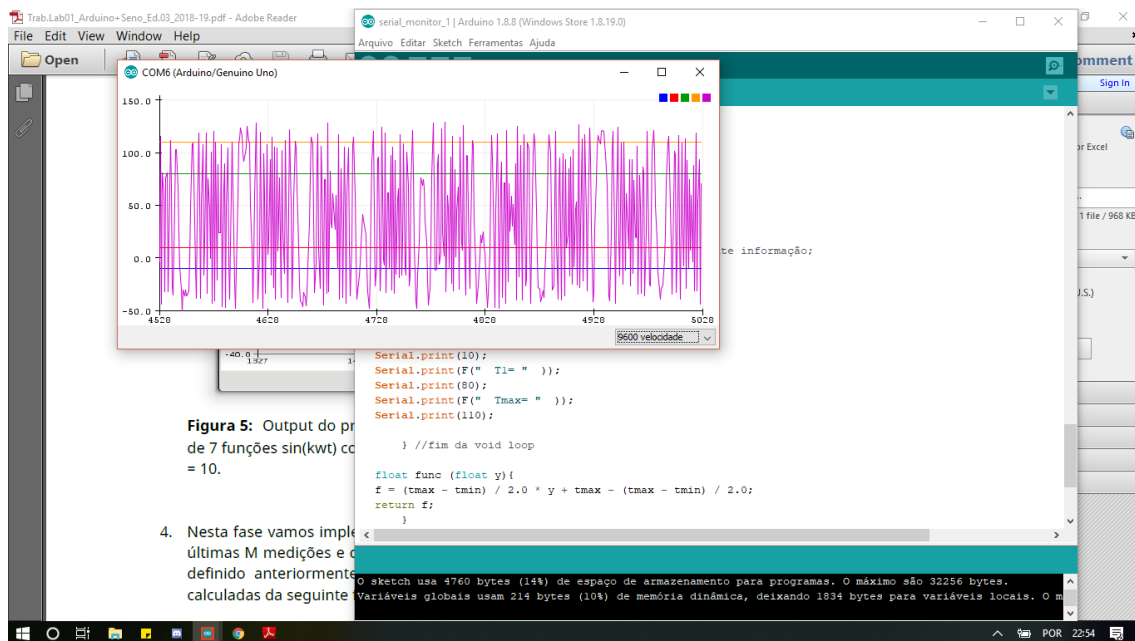
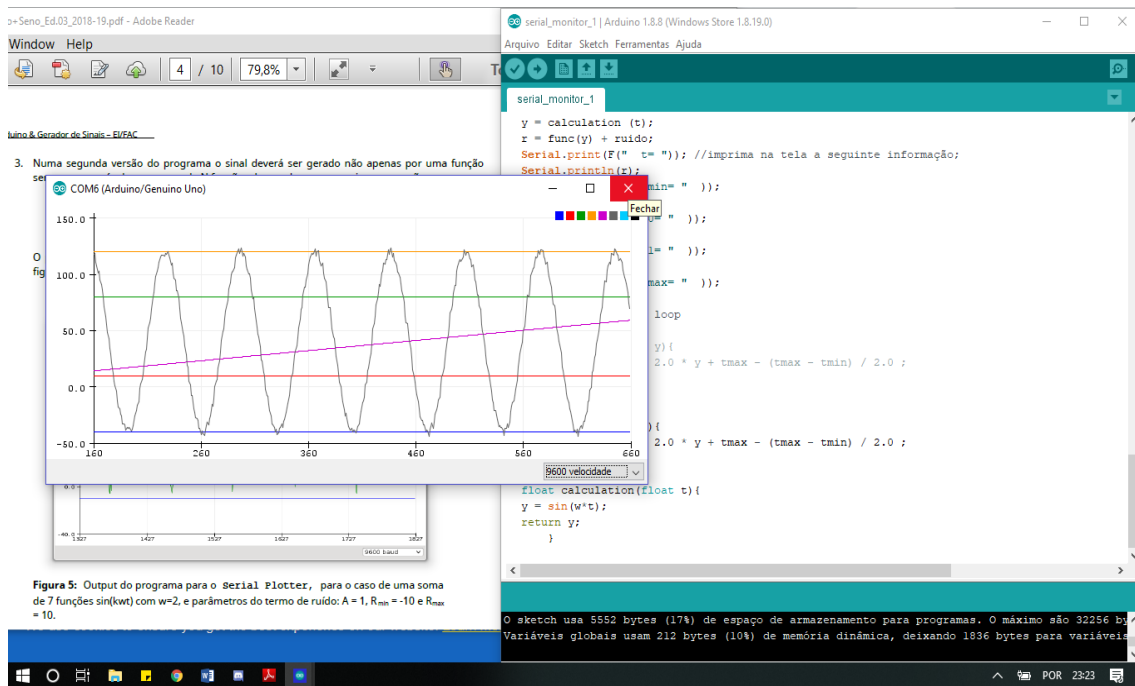


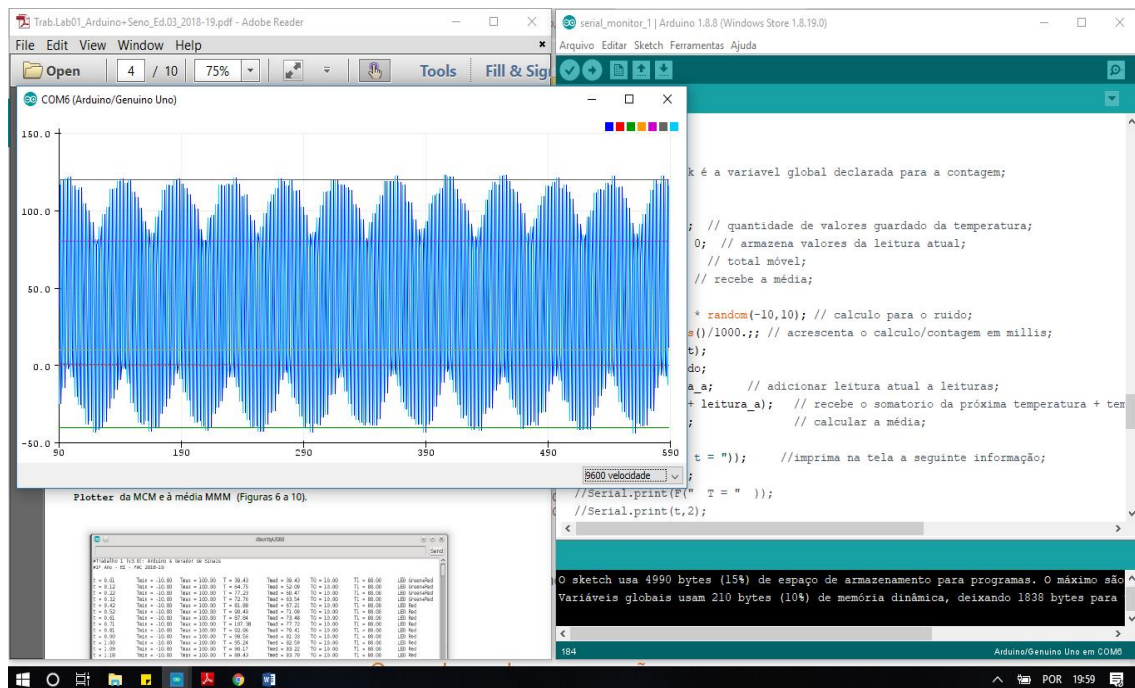
Figura 5: Output do plot de 7 funções $\sin(kwt)$ com $k = 10$.

4. Nesta fase vamos implementar as últimas M medições e o definido anteriormente calculadas da seguinte

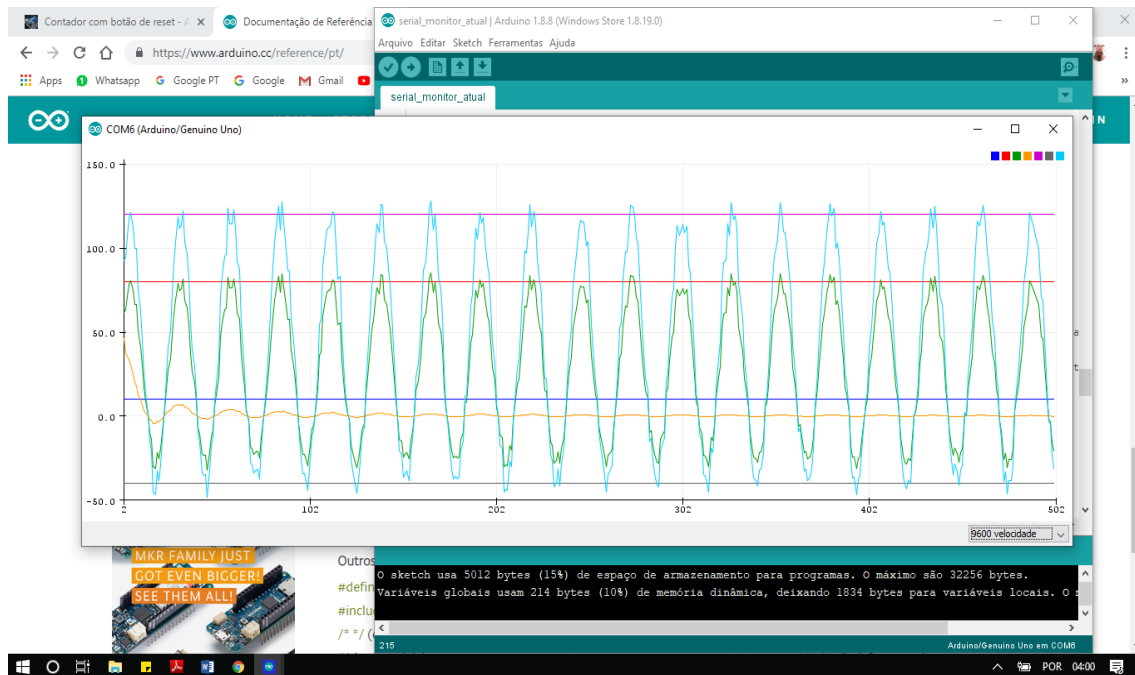
Print teste execução do seno com somador de media – plotter



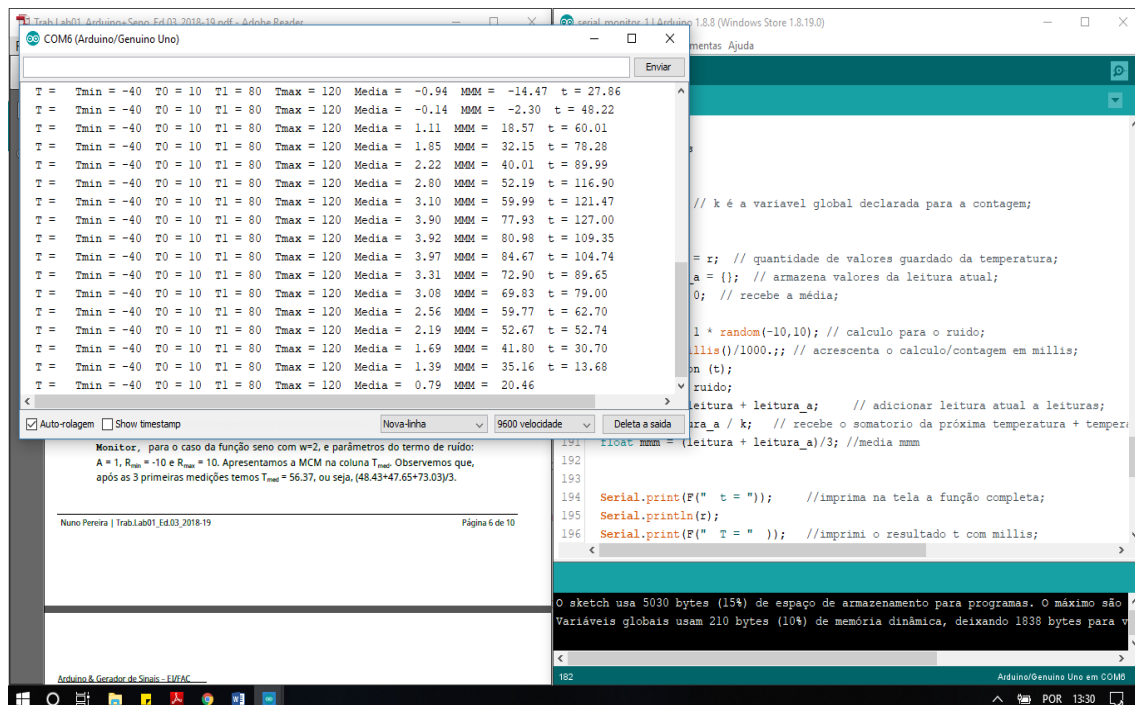
Print teste media Tmed MCM – plotter



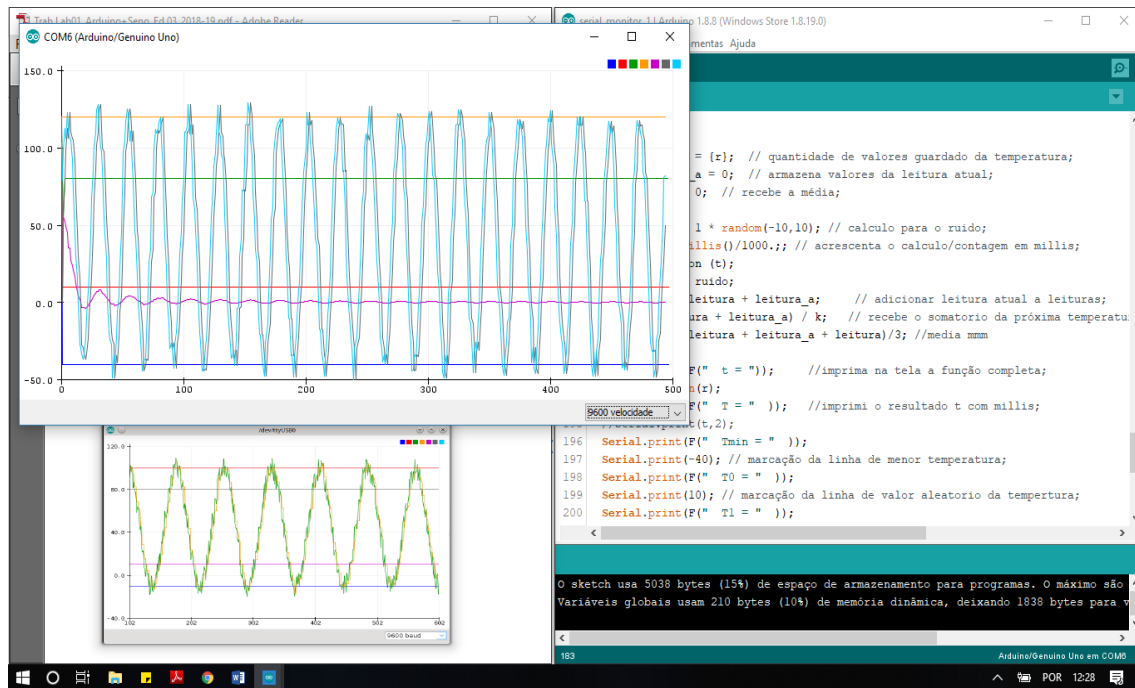
Print teste media acumulativa MMM – plotter



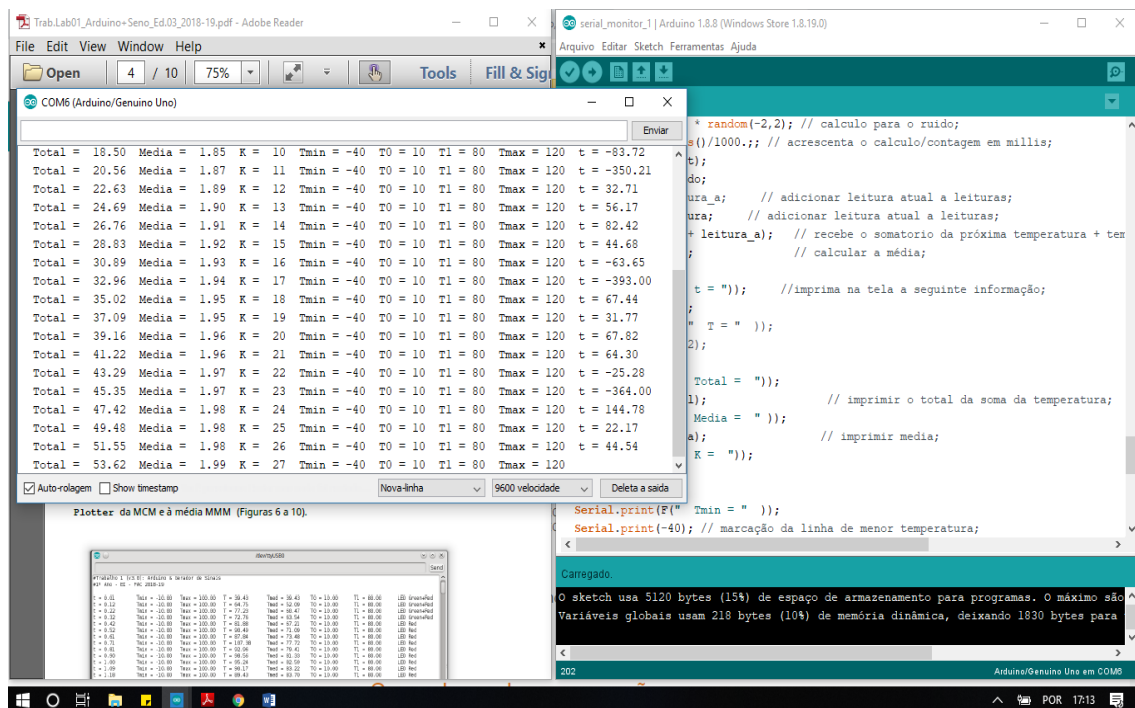
Print teste media MMM – monitor serial



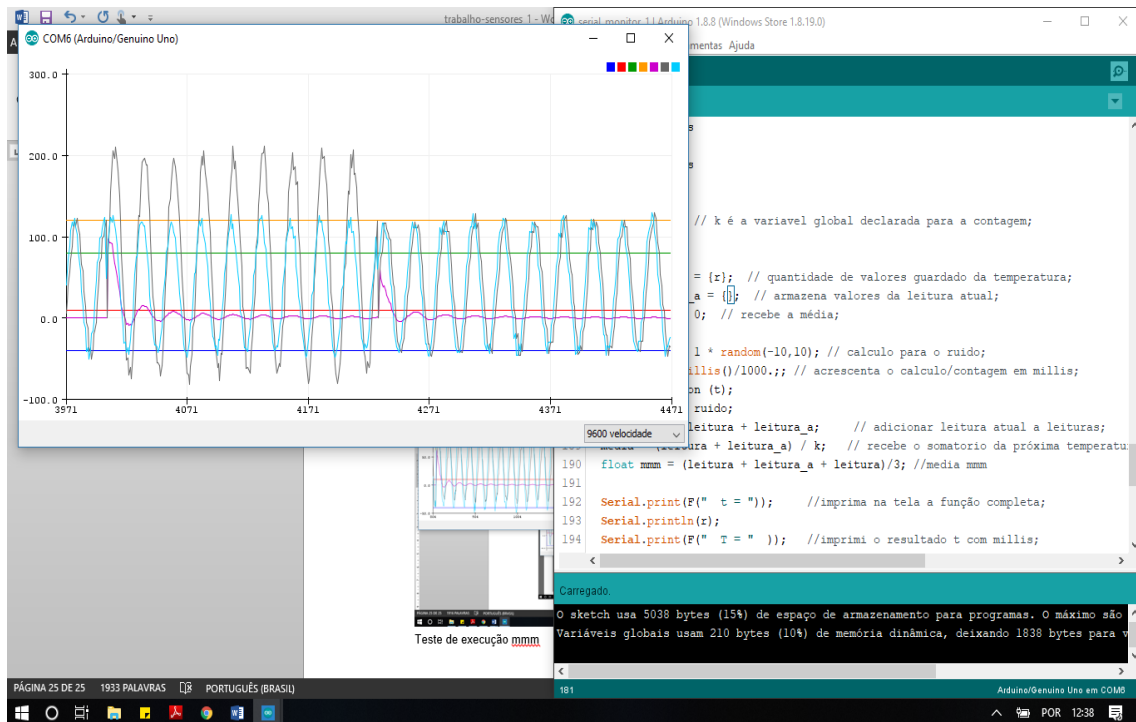
Print teste media MCM corrente – plotter



Print teste media MCM corrente – monitor serial



Print teste execução suavização ou smoothing – plotter



Conclusão

Visando os resultados obtidos concluímos tanto a plataforma Arduino em geral e também a parte dos sensores, led e circuito em si, do programa podem ser usadas em áreas de pesquisas, estudos, aplicação de conhecimentos adquiridos com relevância e relacionadas em pesquisas futuras para um melhor resultado, apesar dos resultados satisfatórios obtidos, sempre podemos continuar melhorando e adequando os códigos de uma maneira mais integra e enxuta assim como ampliar as ações e execução do projeto.

Todos os conceitos estudados nesse projeto podem ser utilizados em outras aplicações e projetos similares, esse estudo utiliza o Arduino, considerado equipamento de baixo custo, torna-se viável e facilmente replicável em projetos futuros.

De uma forma simples e com todos os resultados obtidos, buscamos demonstrar em geral todos os passos executados para a elaboração do projeto, assim como o código e suas impressões em serial monitor, plotter, permitindo assim sua reprodução até com certas modificações de valores, especificações e melhorias tanto na parte de programação e elaboração do circuito.

Concluímos o presente trabalho com o sentimento de ter alcançado resultados satisfatórios, proposto pelo objetivo e manual de elaboração entregue pelo mestre Nuno Pereira e que não existe limite para a aplicação neste modelo de funcionamento e melhoria de qualquer parte do projeto, incluindo com novos modelos de placa e sensores como o de sequência WIFI que esteja relacionado ao desenvolvimento similar do projeto e que o limite das execuções está somente relacionado ao pensamento e conhecimento presente dos seres humanos, conforme várias vezes dito pelo mestre Nuno Pereira. Conforme formos estudando os novos projetos e novos equipamentos e podemos usar essa plataforma Arduino que é muito importante para o controle de vários outros sensores e equipamentos além de criar o conhecimento facilitado com o acoplamento mecânico, robótico e elétrico neste equipamento chamado ARDUINO.

Referências

https://cms.ipbeja.pt/pluginfile.php/192653/mod_resource/content/6/Trab.Lab01_Arduino%2BSeno_Ed.03_2018-19.pdf

hardware: Arduino Uno R3

programa: Arduino IDE

programa: Fritizing

<https://pt.wikipedia.org/wiki/Arduino>

<http://forum.arduino.cc/index.php>

<http://labdegaragem.com/forum/topics>

<http://playground.arduino.cc/Referencia/Extendida>

<https://www.arduino.cc/en/Reference/FunctionDeclaration>

<https://www.arduino.cc/en/Reference/If>

<https://www.arduino.cc/en/Reference/For>

<https://www.robocore.net>

<https://www.arduino.cc/en/Reference/SwitchCase>

<https://www.arduino.cc/en/Reference/While>

<https://www.arduino.cc/en/Reference/DoWhile>

<https://www.arduino.cc/en/Reference/Break>

<https://www.arduino.cc/en/Reference/Continue>

<https://www.arduino.cc/en/Reference/Return>