



Introdução à Engenharia

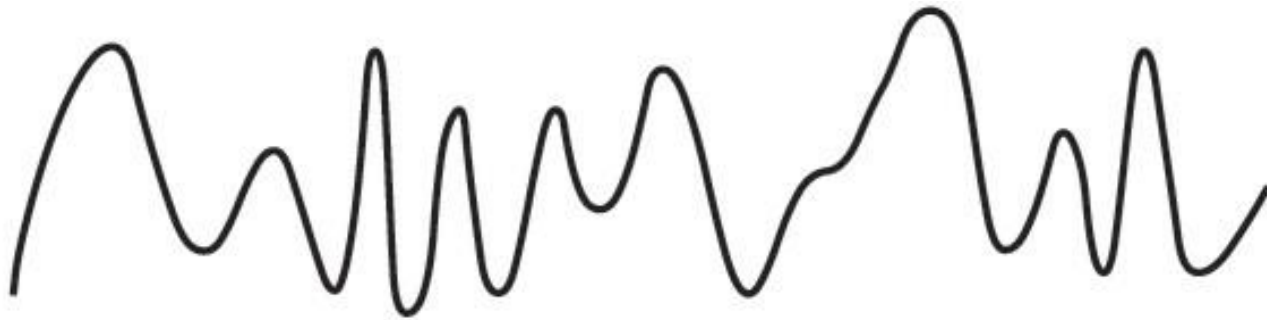
Vinicius Mendonça Martins

viniciusmendonca@gec.inatel.br

Inatel

Sinais Digitais e Analógicos

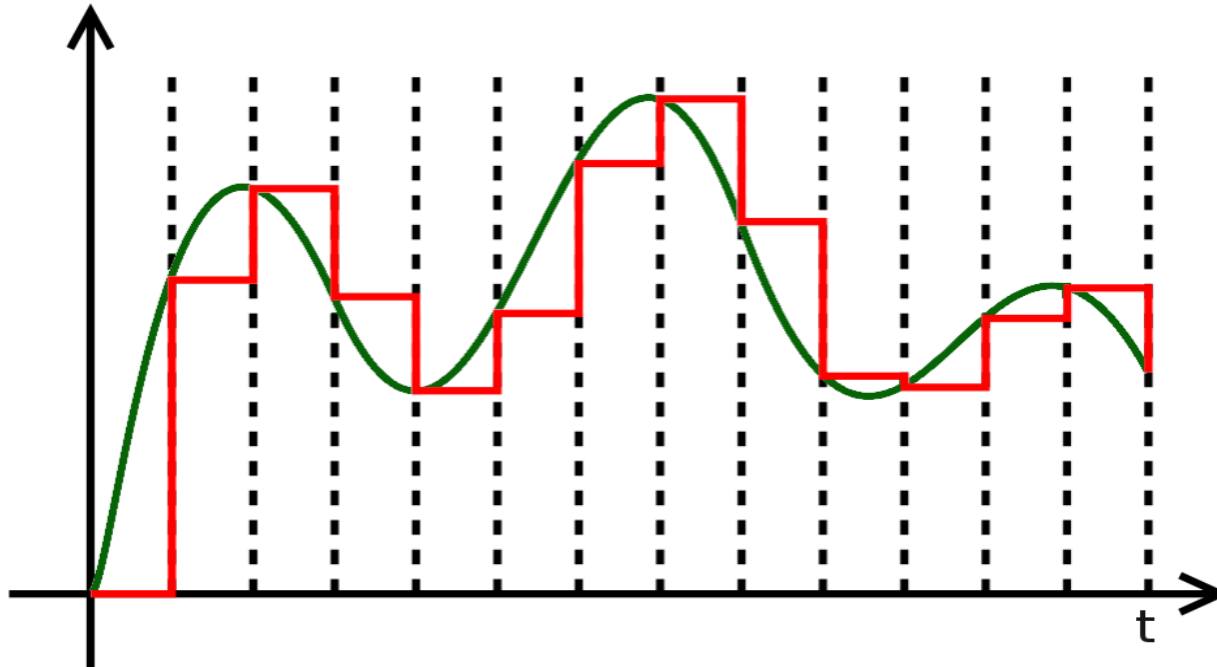
Sinal Analógico



Sinal Digital

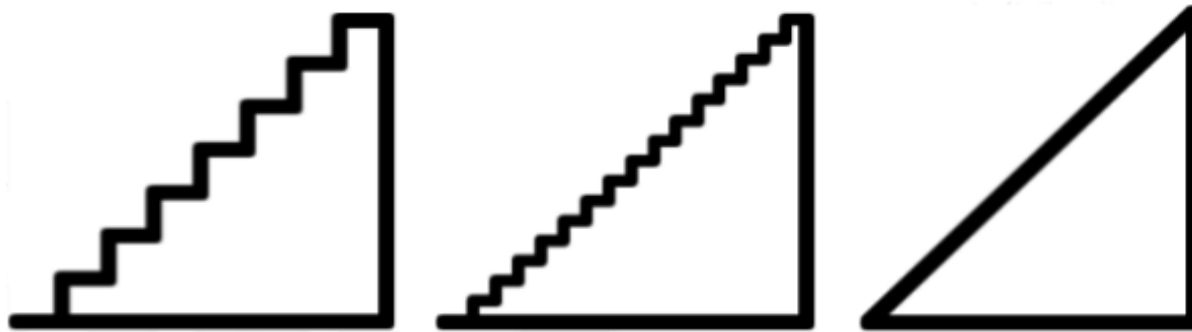


Sinal Discretizado no Tempo



Ao amostrar um sinal (ler seu valor), fazemos isso em um dado intervalo de tempo. Desta forma, estamos discretizando este sinal do tempo.

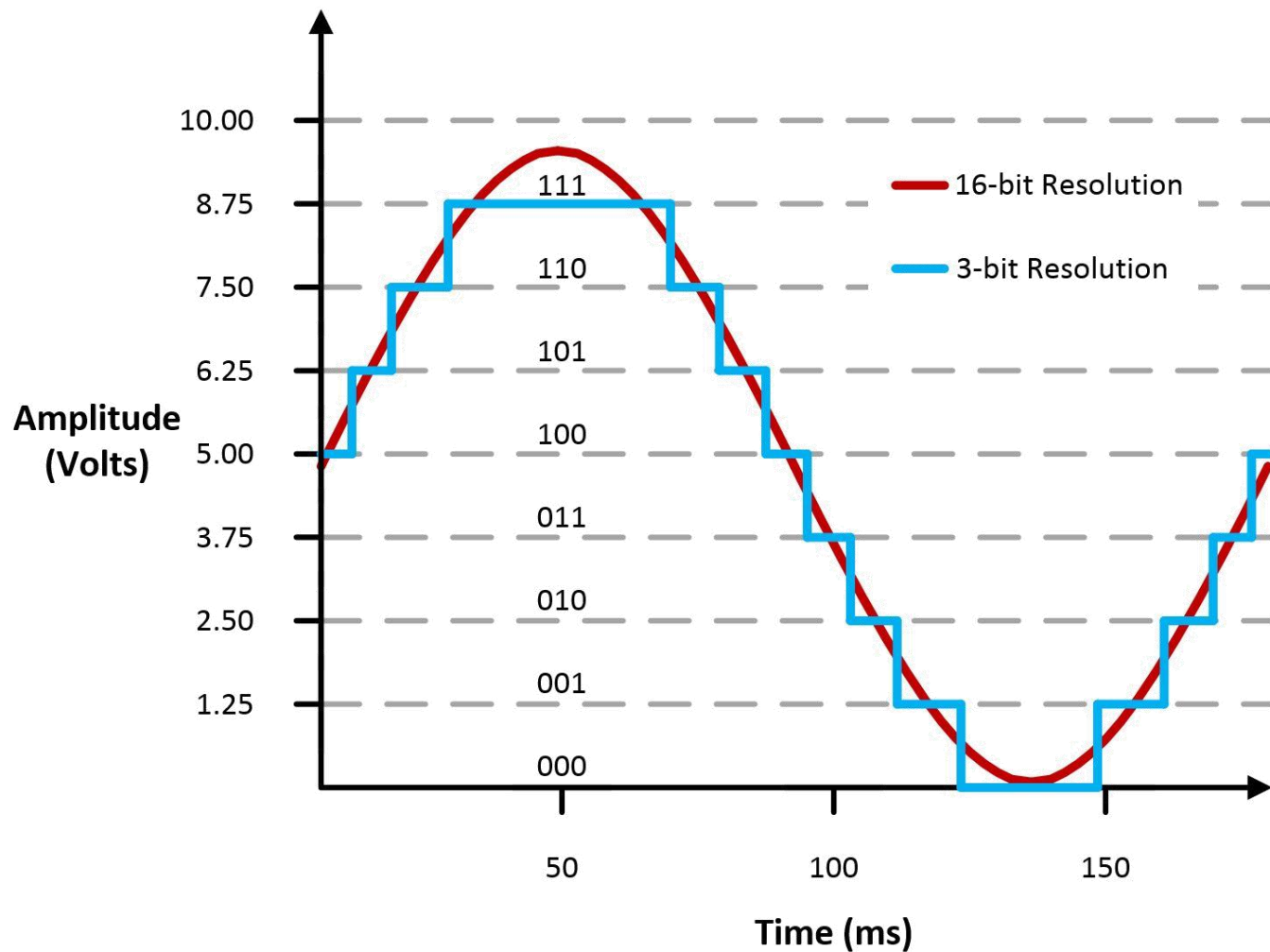
Sinal Analógico Discretizado



Variação da frequência de amostragem na modelagem de um sinal dente de serra.

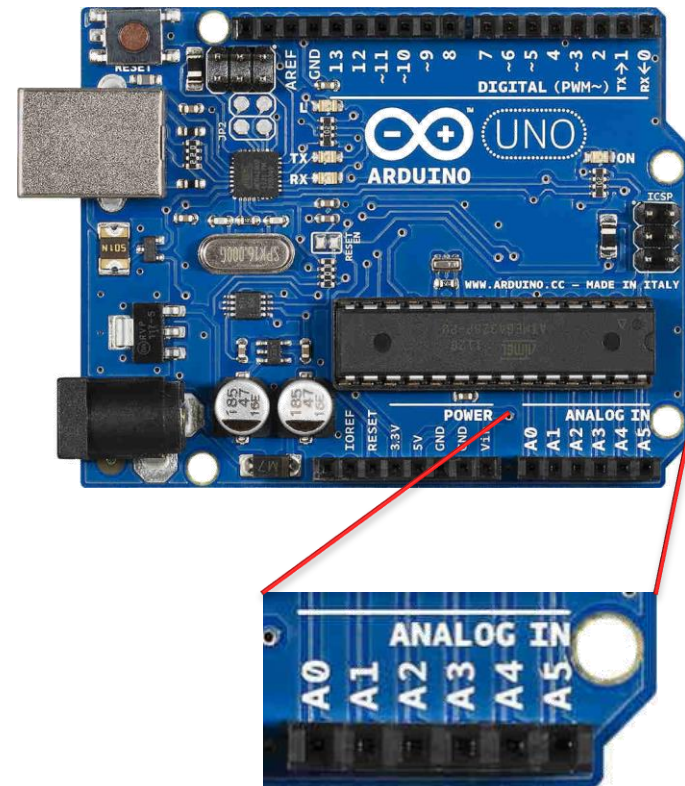
Com o aumento da frequência das amostragens, o sinal modelado passa cada vez mais a se parecer com o sinal original, suavizando os "degraus" existentes.

Sinal Discretizado em Amplitude



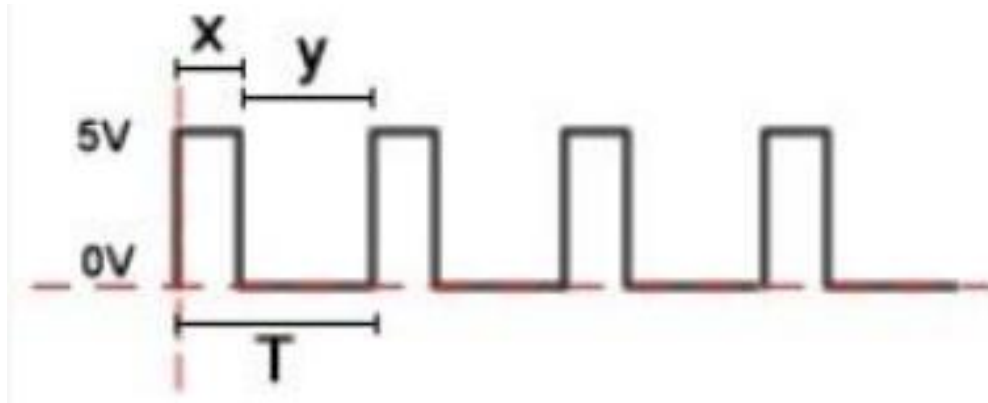
Entradas Analógicas no Arduino

- Pinos: de A0 a A5
- Resolução do ADC: 10 bits
- Range de tensão: 0 a 5V
- Range de valores: 0 a 1023
- Resolução de 4,88mV
$$5V \div 2^{10} = 4,88mV$$
$$5V \div 1024 = 4,88mV$$

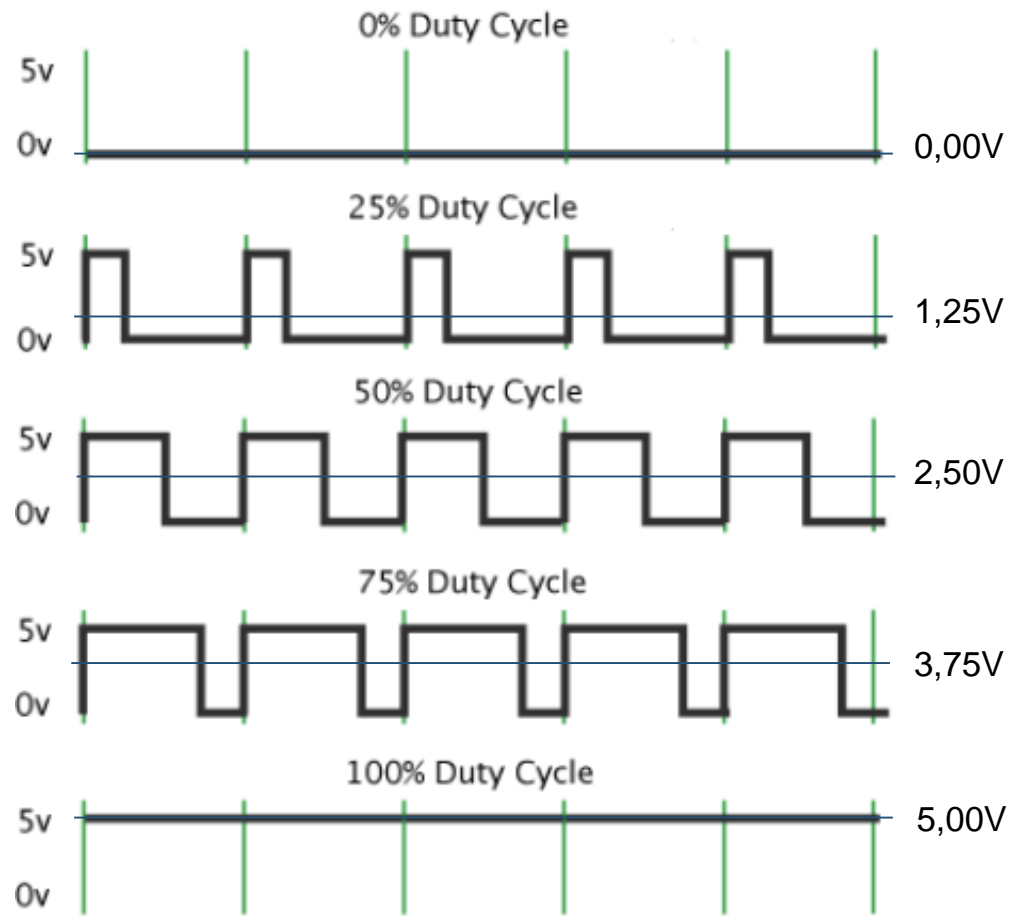


PWM - Pulse Width Modulation

- Técnica para obter resultados analógicos por meios digitais.
- O Duty Cycle é a razão do tempo em que o sinal permanece na tensão máxima (5V no Arduino) sobre o tempo total de oscilação.

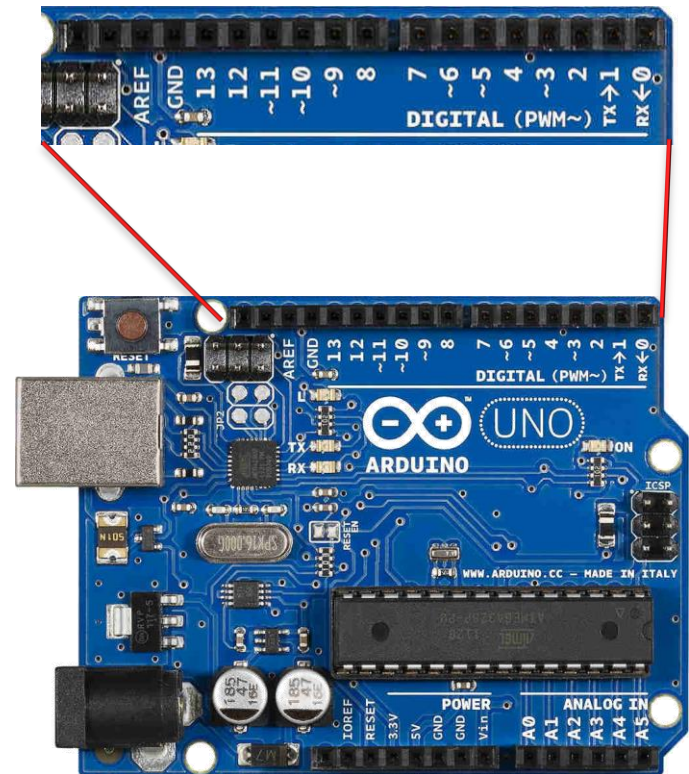


PWM no Arduino



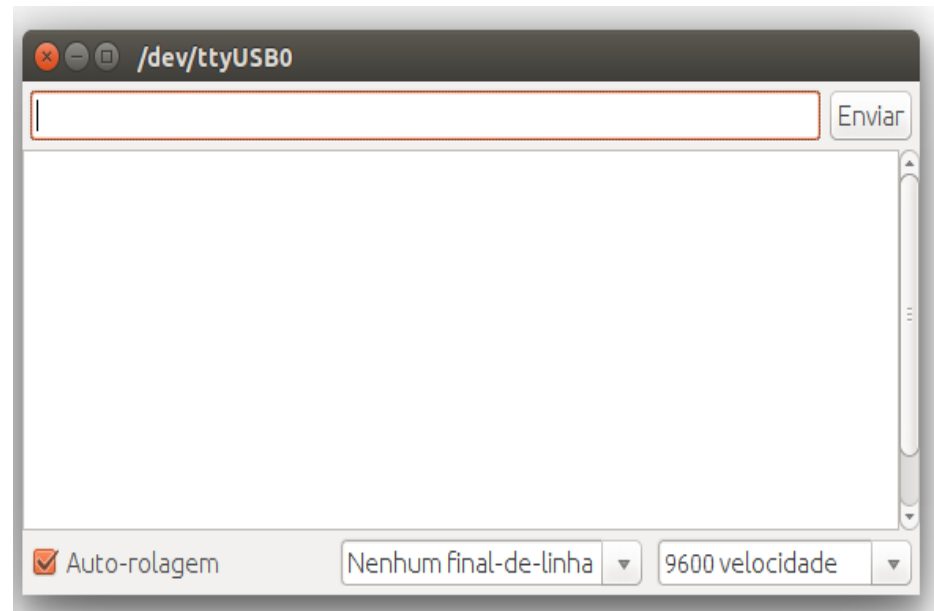
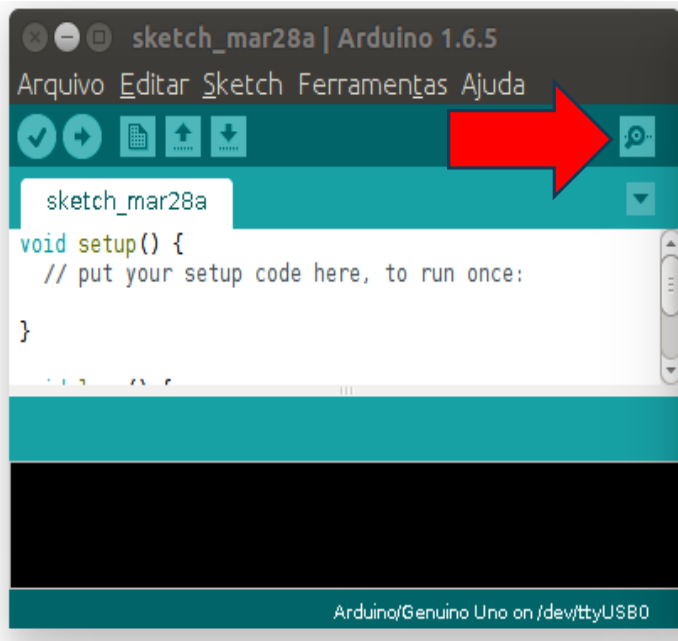
PWM no Arduino

- Pinos (digitais): 3, 5, 6, 9, 10 e 11
- Resolução do PWM: 8 bits
- Range (tensão): 0 a 5V
- Range (valores): 0 a 255]
- Resolução de 19,53mV
$$5V \div 2^8 = 19,53mV$$
$$5V \div 256 = 19,53mV$$



Serial Monitor

Interface gráfica que facilita a comunicação entre o Arduino e um computador



Funções Importantes

- `Serial.begin(9600);`
`/*Inicia porta serial e define a velocidade de transmissão (9600bps) */`
- `Serial.println(x);`
`/* imprime o valor de x na porta serial */`
- `char c = Serial.read();`
`/* c recebe caracter da porta serial */`

Funções Importantes

- `X = analogRead(A0);`
// x recebe o valor do pino (de 0 a 1023)
- `analogWrite(6,127);`
// aplica PWM no pino 6 (Duty Cycle = $127/255 = 50\%$)
// se Duty Cycle = 50%, tensão = 50% de 5V = 2.5V
- `novo_valor = map(valor,0,1023,0,200);`
// escala de "valor" = 0 a 1023
// escala de "novo_valor" = 0 a 200

Material de Apoio

Livros:

<https://goo.gl/KLe7p6>

Sites:

www.filipeflop.com/blog

www.arduinoecia.com.br

www.instructables.com, buscar por "arduino"

Canais de YouTube:

All Electronics: <https://goo.gl/KQfjsX>

WR Kits: <https://goo.gl/6v1NNe> (Mais de 130 exemplos)

Exercício (parte 1)

Elabore um programa que, de acordo com o valor de tensão lido em um potenciômetro, envie valores específicos de tensão para um LED. Isso deve ser feito de acordo com a tabela a seguir:

Intervalo de V de entrada (potenciômetro)	Valor de V de saída (LED)
0V a 1V (inclusive)	0V
1V a 2V (inclusive)	1V
2V a 3V (inclusive)	2V
3V a 4V (inclusive)	3V
4V A 4.5V (inclusive)	4V

Exercício (parte 2)

Caso a entrada seja superior a 4.5V, o led deverá ser apagado, um buzzer acionado e o sistema deverá travar. O buzzer só será desligado caso o Arduino receba o caracter '0' pela Serial.

Os valores da tensão de entrada e da tensão de saída devem ser constantemente mostrados no Serial Monitor enquanto o sistema estiver funcionando (não estiver travado).

Extra:

- Mostrar também o valor de Duty Cycle (em %);
- Acionar o buzzer como uma sirene.

Dicas

Cálculo do valor correspondente às tensões especificadas: Lembre-se que a entrada analógica possui uma resolução de 10 bits (valores de 0 ~ 1023) e a saída por PWM possui uma resolução de 8 bits (valores de 0 ~ 255);

Travamento do sistema: utilize um "while" para monitorar o valor lido da porta serial;

Dúvidas? Me mande um e-mail e/ou me procure no horário de atendimento.

Entrega de Exercício

- **Entrega por e-mail:**

- Destinatário: viniciusmendonca@gec.inatel.br
- Assunto: [C201 – L#] - Entrega 02
 - Substitua # pelo número da sua turma.
- Corpo: Deve conter Nome e Matrícula da dupla.

- **Data de entrega:**

- A entrega deve ser feita até **25/03/2019**, às 23:59.