



Amplificação Lock-In para PPG

Grupo:

- Brunna de O. Rodrigues;
- Gabriel E. P. L. e Silva;
- Guilherme A. N. de Barros.



Conteúdo

01

Sinal de Interesse

Fotopletismografia (PPG)

02

Amplificação


Diagrama de Blocos

Lock-In

03

Resultados

Descrição do
funcionamento





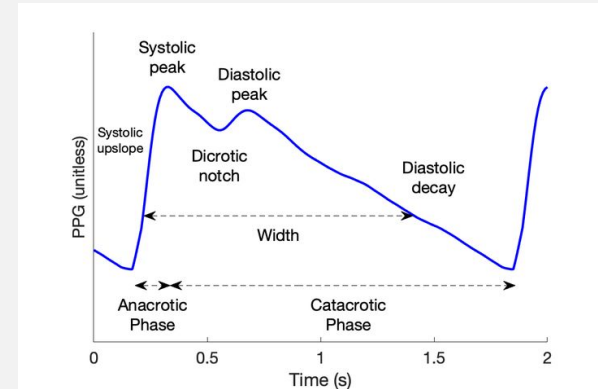
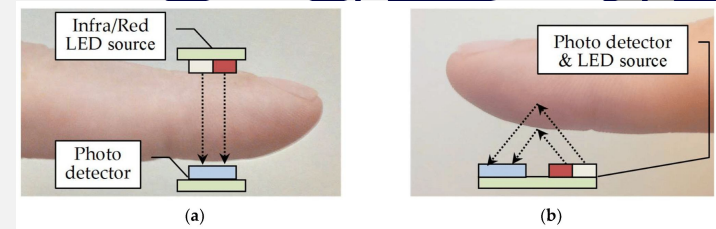
01

Sinal de Interesse

Fotopletismografia (PPG)

Sinal de Interesse

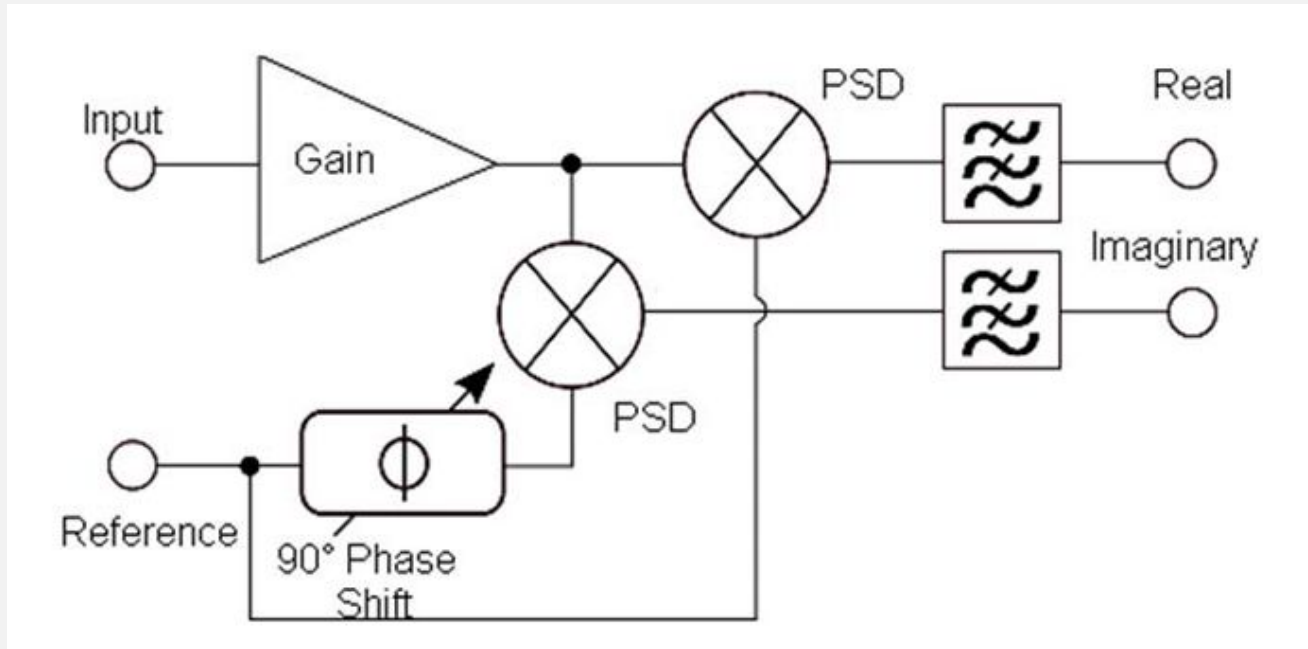
A **fotopletismografia (PPG)** é uma técnica óptica não invasiva utilizada para detectar variações no volume sanguíneo no leito microvascular dos tecidos. Isso é realizado iluminando a pele com luz, geralmente proveniente de LEDs, e medindo as mudanças na absorção ou reflexão dessa luz, que ocorrem devido às oscilações no fluxo sanguíneo durante cada ciclo cardíaco.





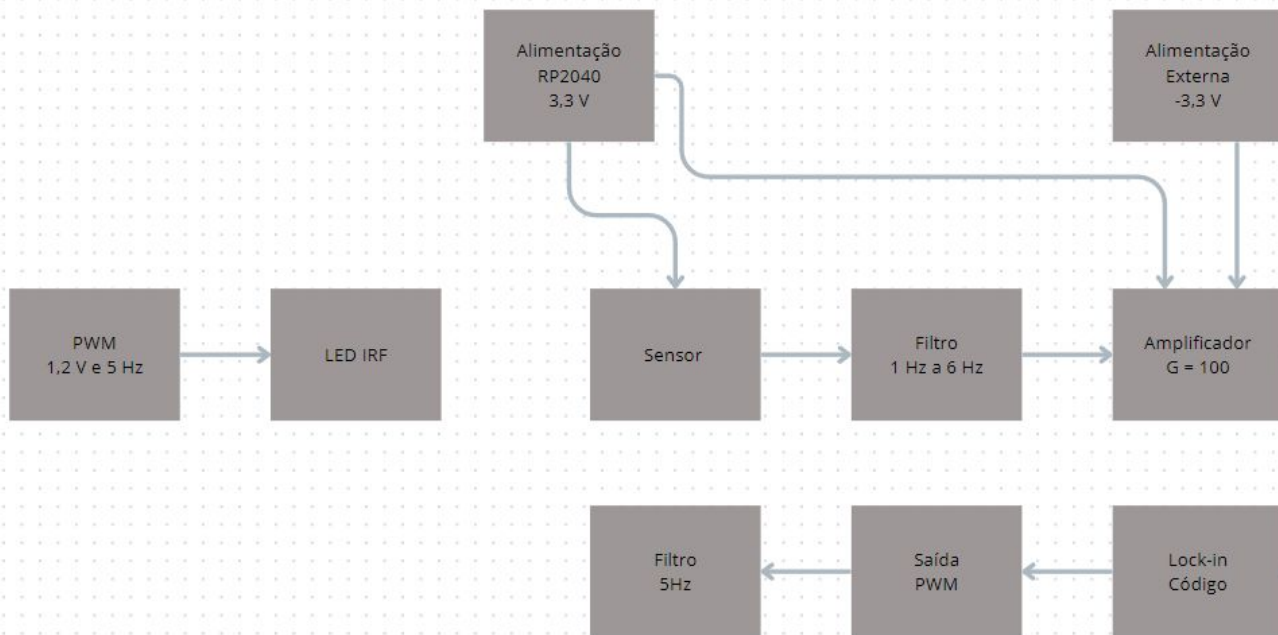
02

**Amplificação
Lock-In**

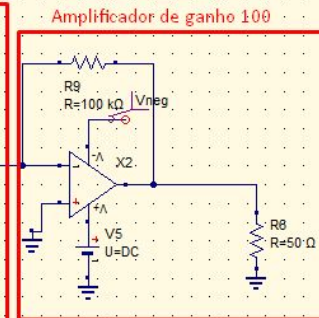
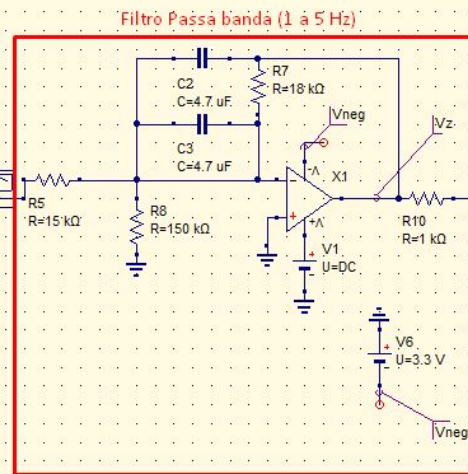
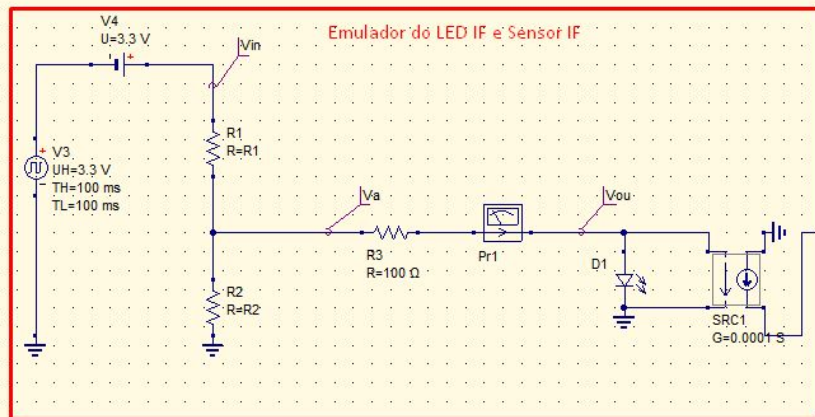


Esquemático básico de um amplificador lock-in

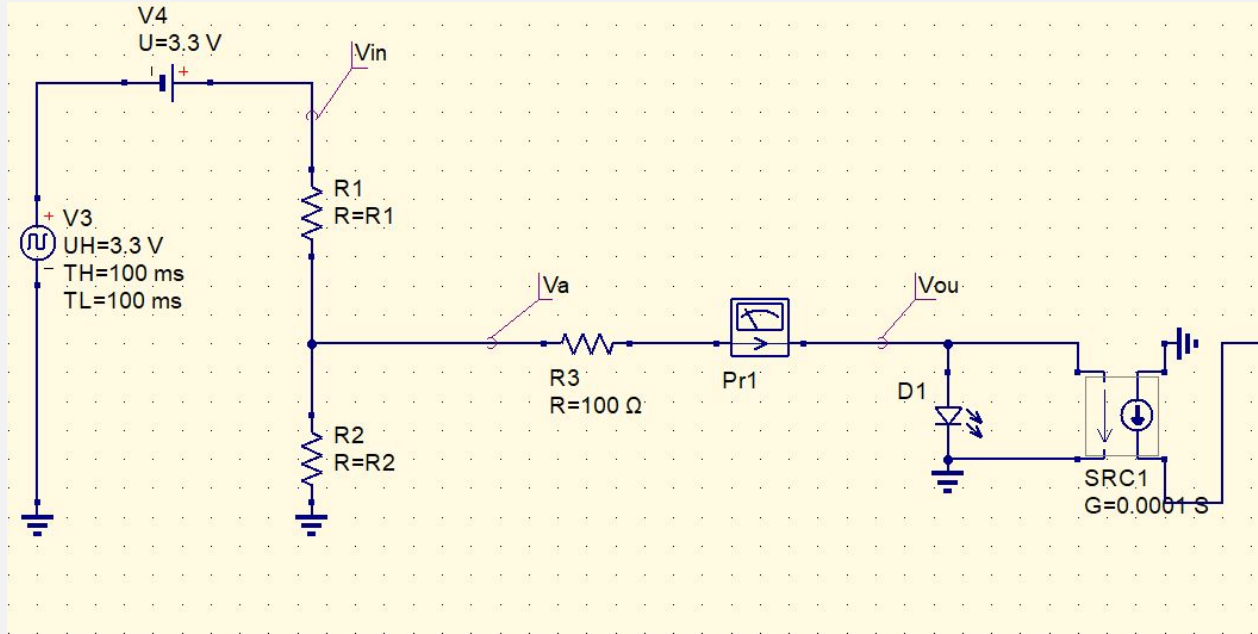
Diagrama de blocos



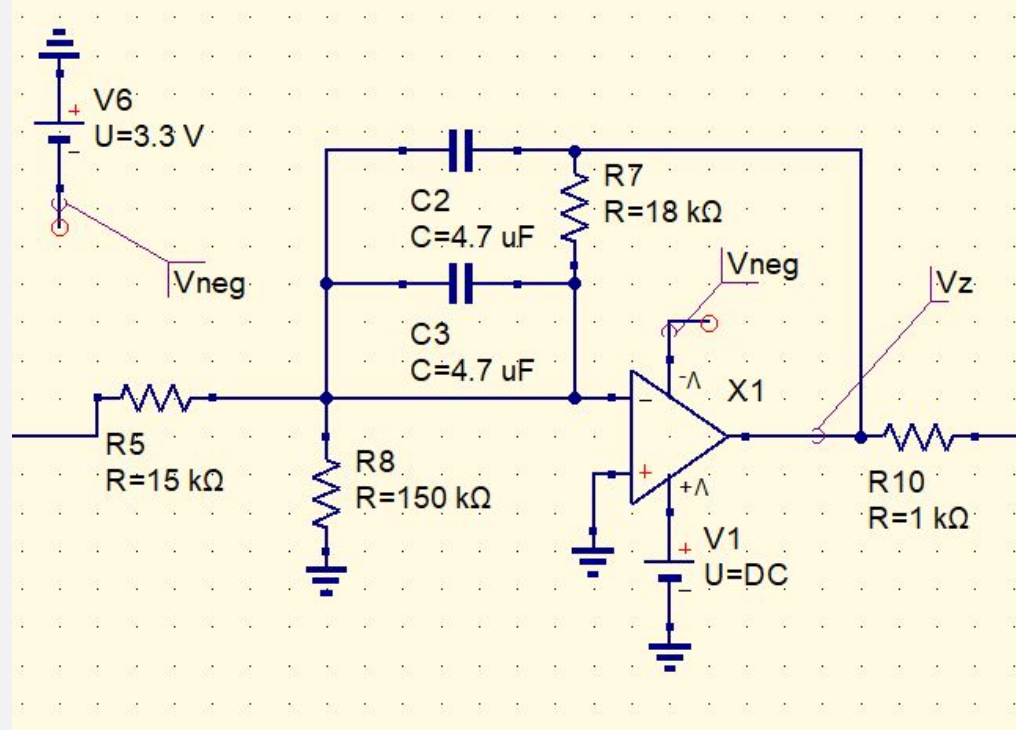
Circuito Analógico - QucsStudio



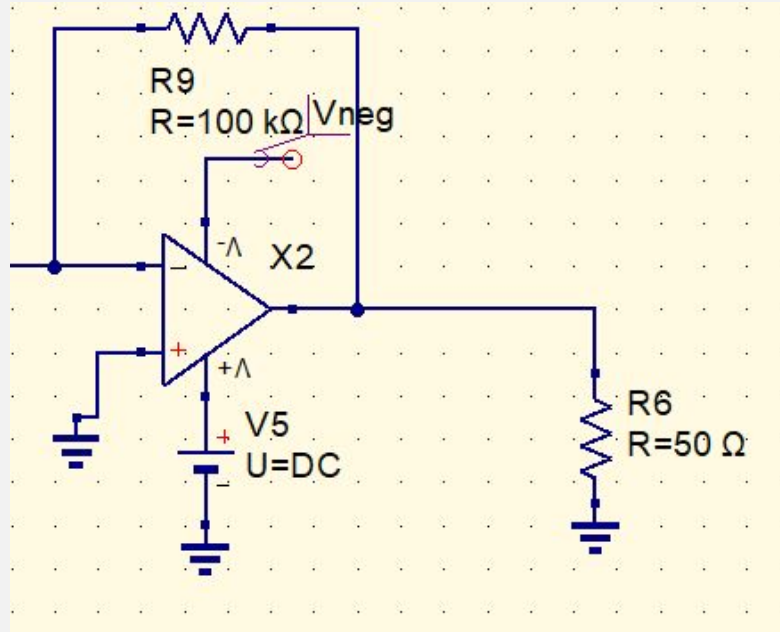
Circuito Analógico - QucsStudio



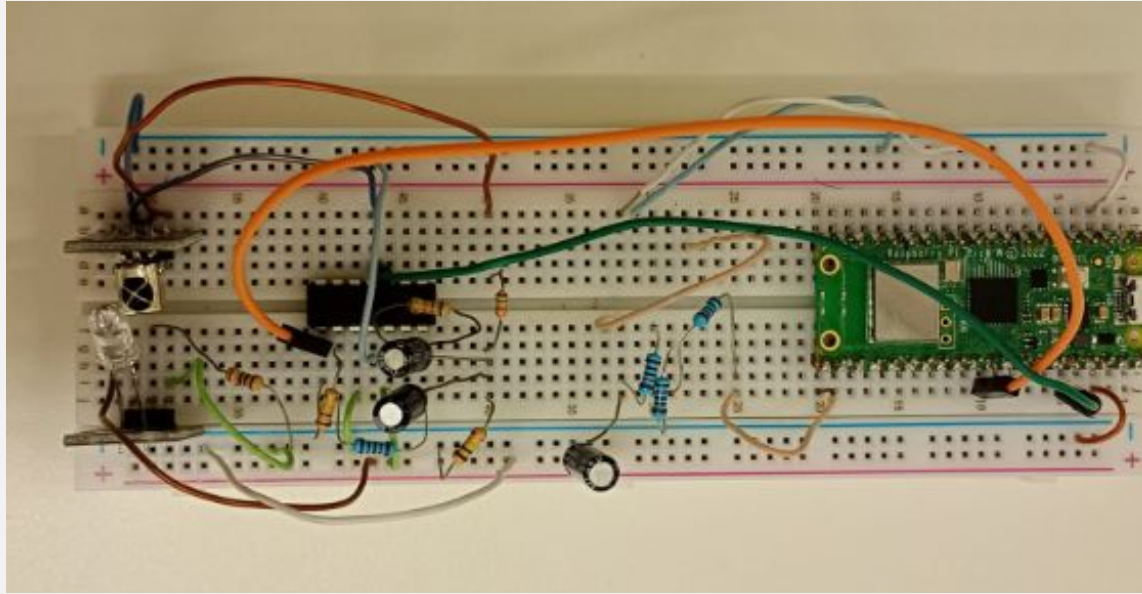
Circuito Analógico - QucsStudio



Circuito Analógico - QucsStudio



Circuito Analógico - Montagem



Lock-in - Código

```
# Função principal que implementa o algoritmo de lock-in
def main_lock_in():
    global previous_input_value, signals

    # Leitura e pré-processamento do sinal de entrada
    x = 0.5 + 0.6 * read_adc() # Escala e desloca o valor do ADC
    filtered_input = low_pass_filter(x, previous_input_value) # Aplica o filtro passa-baixa
    signals = signals + [filtered_input] # Adiciona o valor filtrado à lista de sinais

    # Garante que a lista de sinais não exceda o número de amostras
    if len(signals) > samples:
        signals.pop(0) # Remove o valor mais antigo

    # Calcula o valor médio do sinal
    average_signal = calculate_average_signal(signals)

    # Calcula o valor de lock-in
    lock_in_value = lock_in_detection(filtered_input, average_signal)
```

Lock-in - Código

```
# Calcula o duty cycle para o PWM com base no valor de lock-in
duty_cycle = round(amp * (1 + lock_in_value)) # Ajusta a escala para o PWM
pwm_out.duty_u16(duty_cycle) # Aplica o duty cycle no PWM

# Exibe o valor de lock-in no console
print(f"Valor Lock-in: {lock_in_value:.4f}")

# Aguarda o próximo intervalo baseado na frequência de referência
time.sleep(intervalo_ref_initial / samples)

# Atualiza o valor filtrado anterior para a próxima iteração
previous_input_value = filtered_input

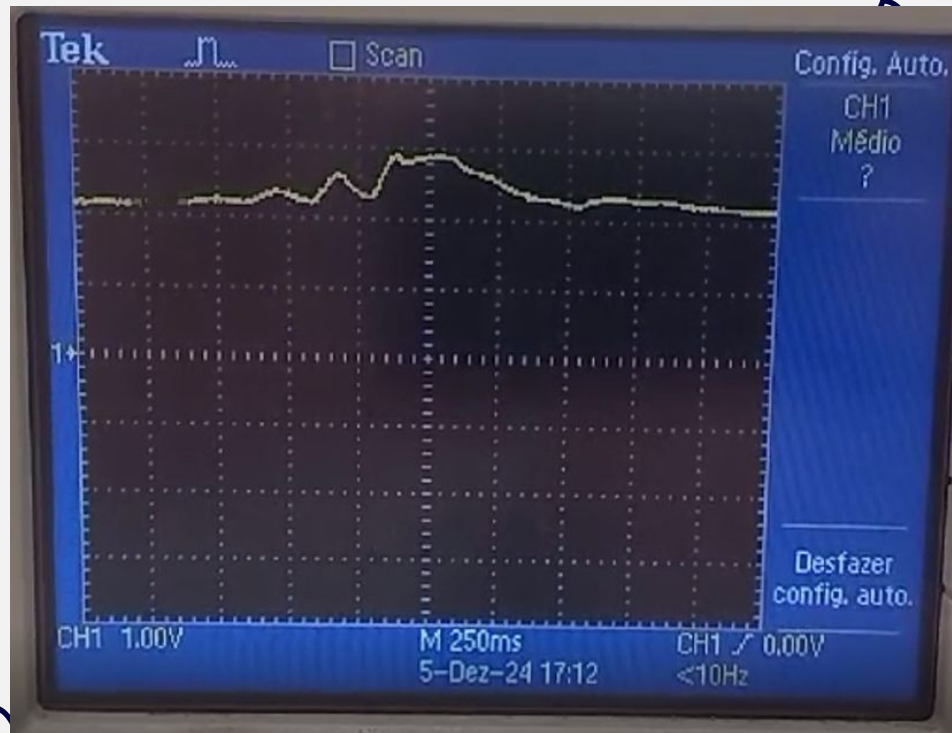
# Retorna o valor calculado de lock-in
return lock_in_value
```



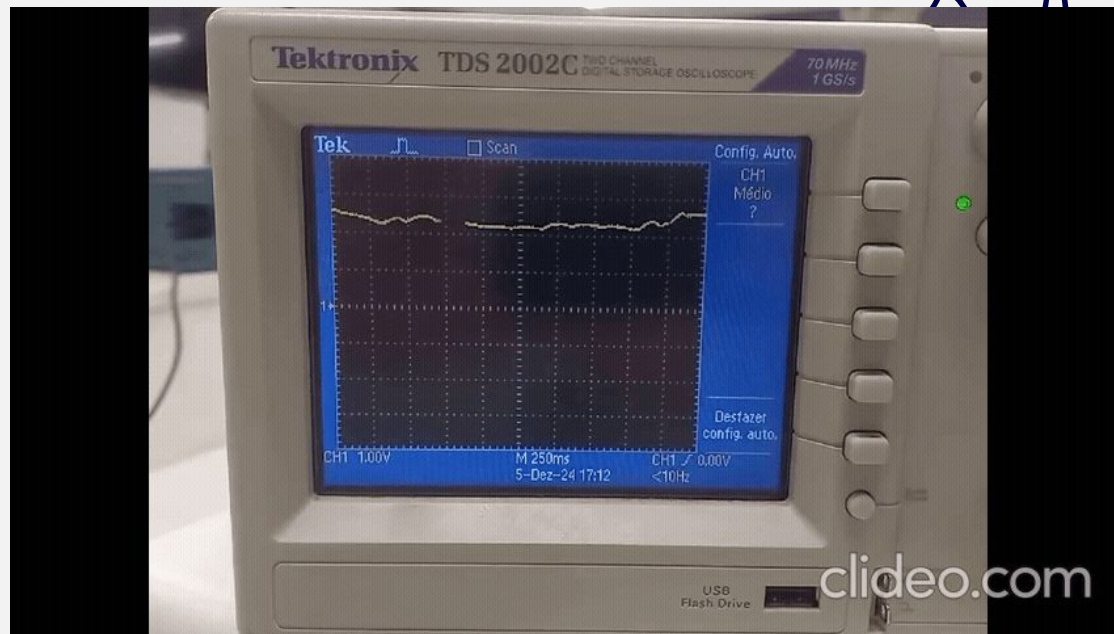
03

Resultados

Resultado



Resultado



clideo.com