Medidor de impedâncias utilizando amplificador *lock-in*

Aquilles Santana & Sofia Schneider

Departamento de Engenharia Elétrica e Eletrônica

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

O que é *lock-in*?

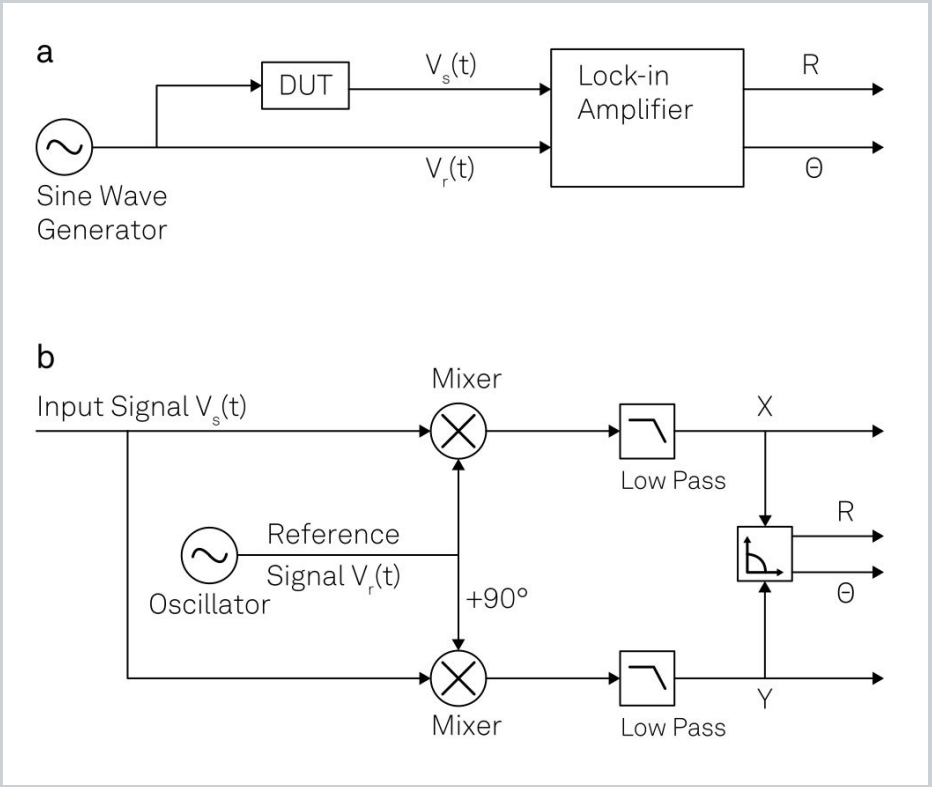
O **amplificador *lock-in*** utiliza uma técnica conhecida como *phase-sensitive detection* (detecção sensível à fase) para extrair uma componente de um sinal numa frequência e fase específicas, definidas por um sinal de referência.

Isso permite que o amplificador tenha um bom desempenho mesmo com tensões de entrada muito baixas e com ruído considerável.

Como funciona um *lock-in*?

O sinal de entrada é multiplicado com o sinal de referência e passa por um filtro. O resultado equivale a parte real do sinal.

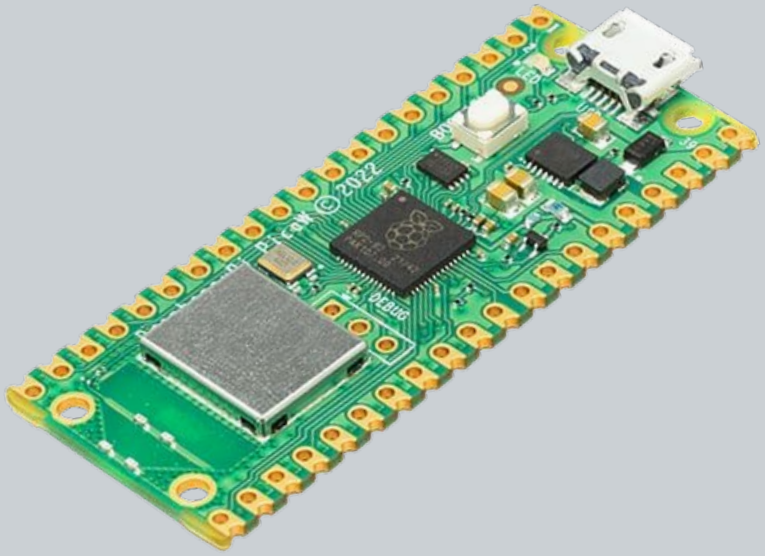
Para obter a parte imaginária, realiza-se o procedimento com uma defasagem de 90 graus no sinal de referência.



Objetivo do projeto

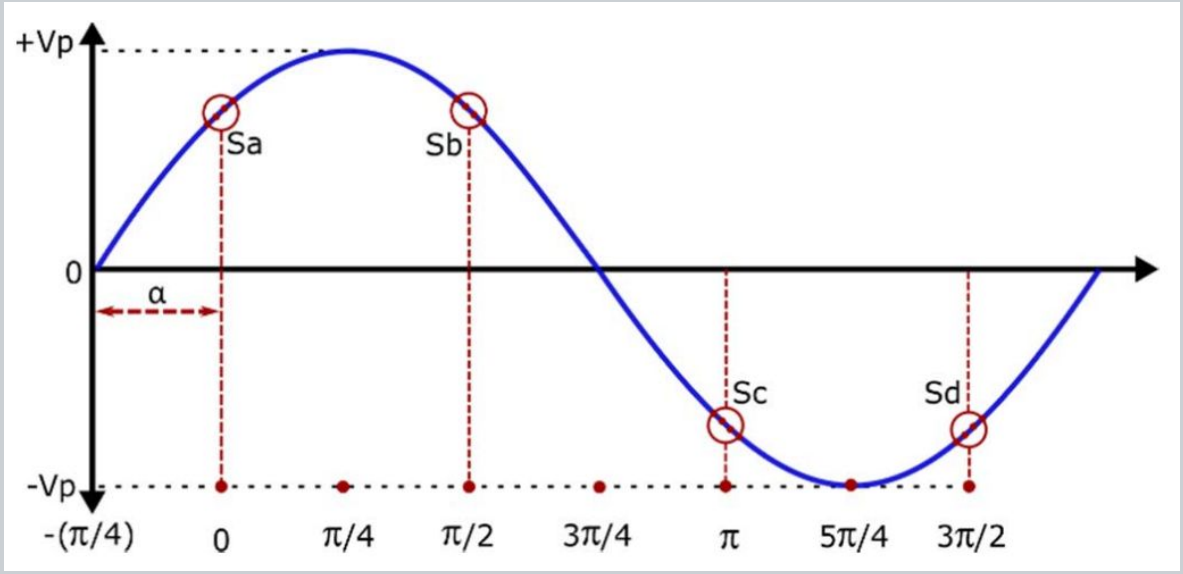
Desenvolver, utilizando a técnica do amplificador *lock-in* e um Raspberry Pi Pico W (ou outro microcontrolador genérico), um medidor de impedâncias com erro relativo inferior a 10%.

Para reduzir ao máximo possível a quantidade de componentes utilizados, foi decidido implementar o *lock-in* digital, onde as operações são feitas pelo microcontrolador.



Método alternativo de *lock-in*

Como a operação de multiplicação é muito custosa em microcontroladores, utilizou-se um método alternativo: ao invés de realizar a multiplicação dos sinais, é realizada a subtração entre 4 amostras obtidas a cada 90 graus do sinal de saída.



Método alternativo de *lock-in*

A parte real é calculada subtraindo a amostra de 90° com a de 270°, e a parte imaginária subtraindo a amostra de 0° com a de 180°. O coeficiente *α* é a diferença entre o *zero-crossing* da referência e da saída.

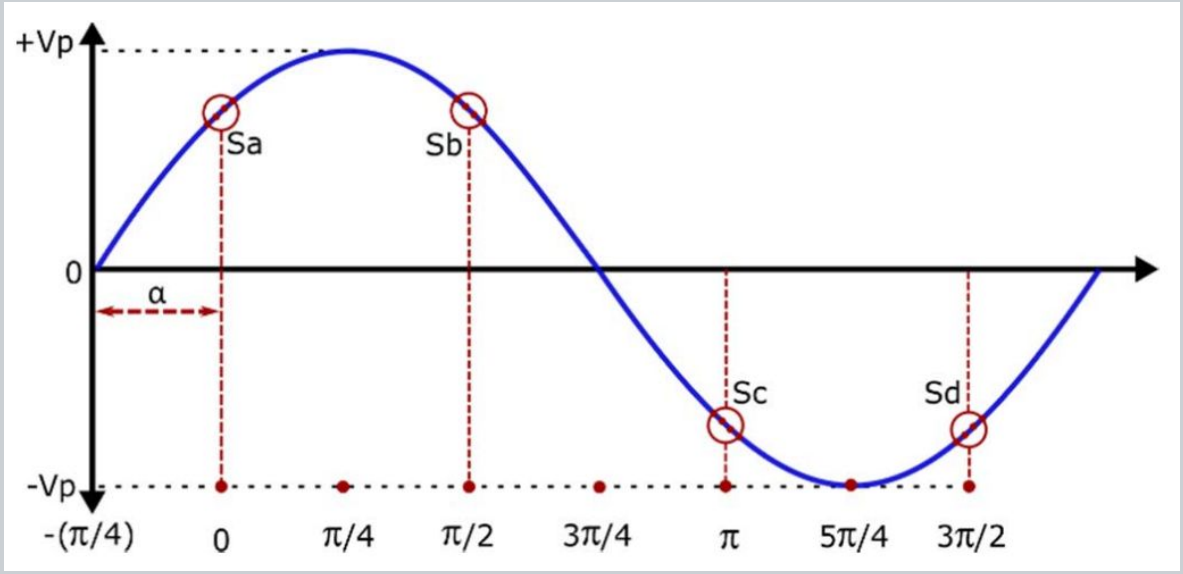
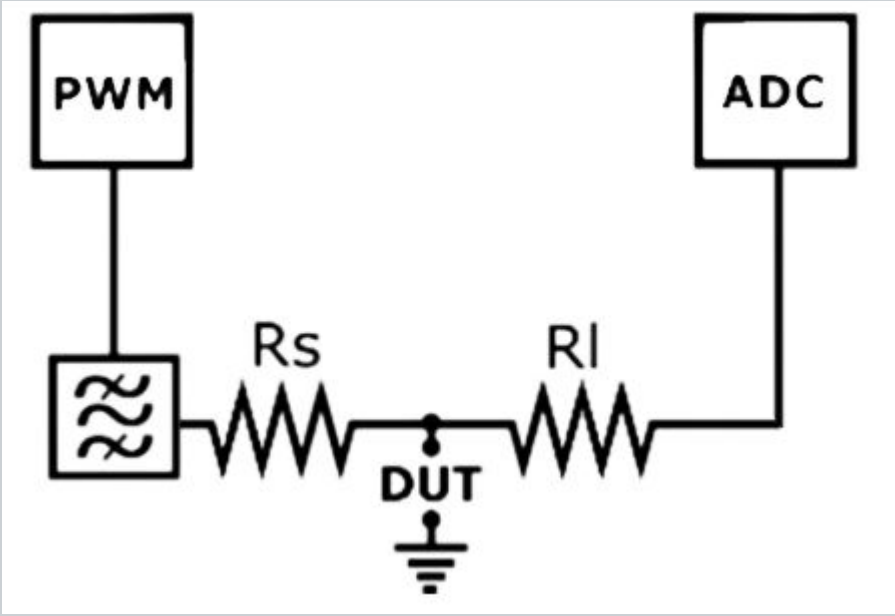


Diagrama de blocos

Para gerar o sinal de referência, utiliza-se de um gerador PWM do microcontrolador, na frequência de 500 Hz, que então passa por um filtro passa-banda ativo de segunda ordem.

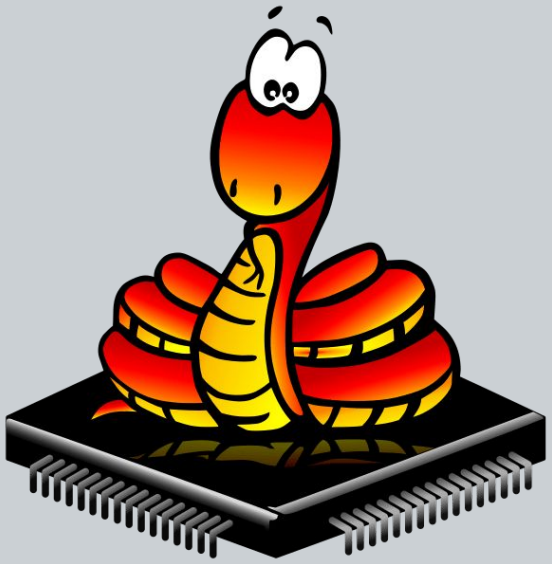
O resistor R_S (definido como 10 kΩ), limita a corrente em caso de DUT's pequenos. O resistor RI é a própria impedância do ADC.



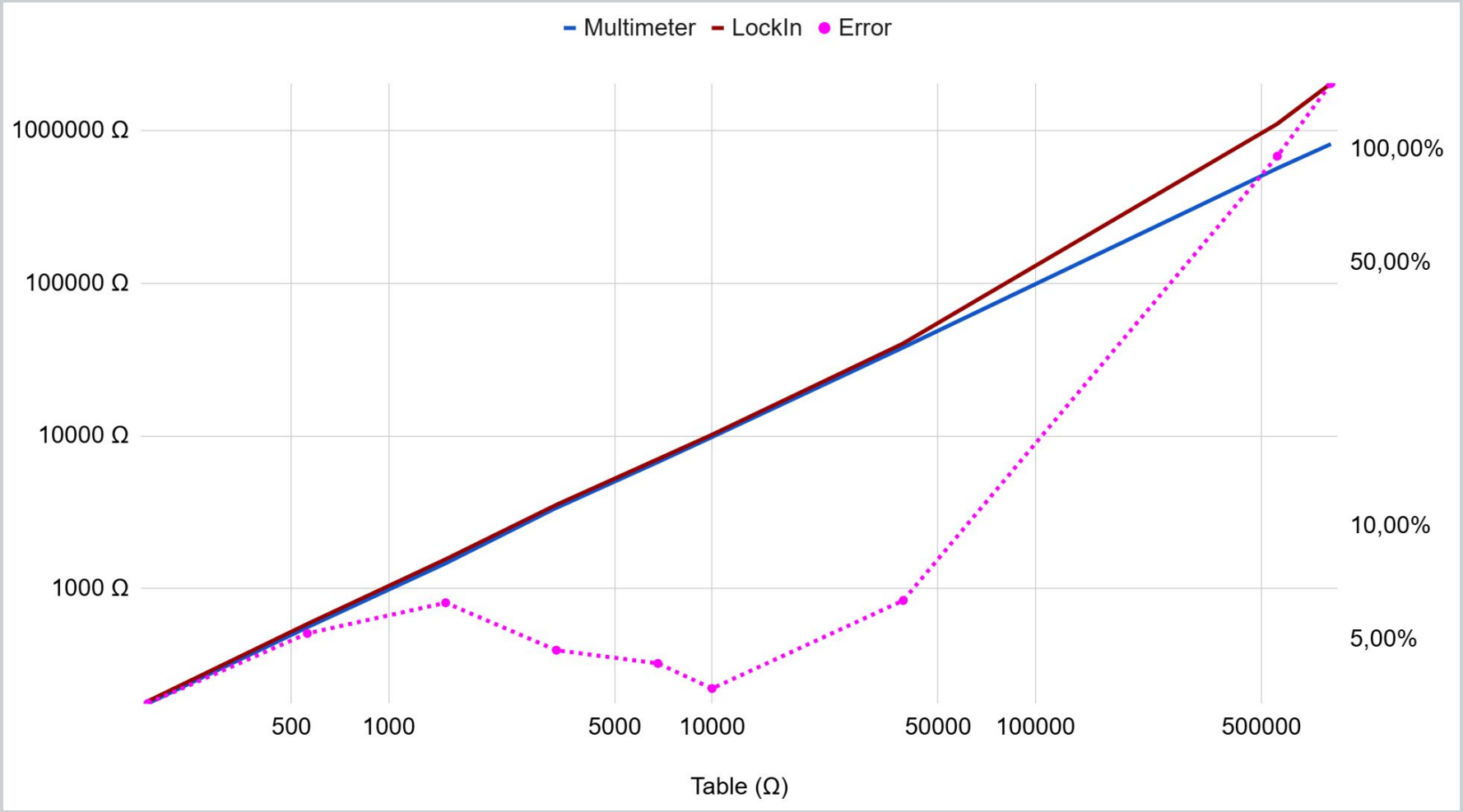
Limitações do MicroPython

Durante o desenvolvimento, notamos que a medida de um valor do ADC demorava muito para nossa aplicação, em torno de 13 µs. Isso fazia com que a determinação do *zero-crossing* exato fosse difícil.

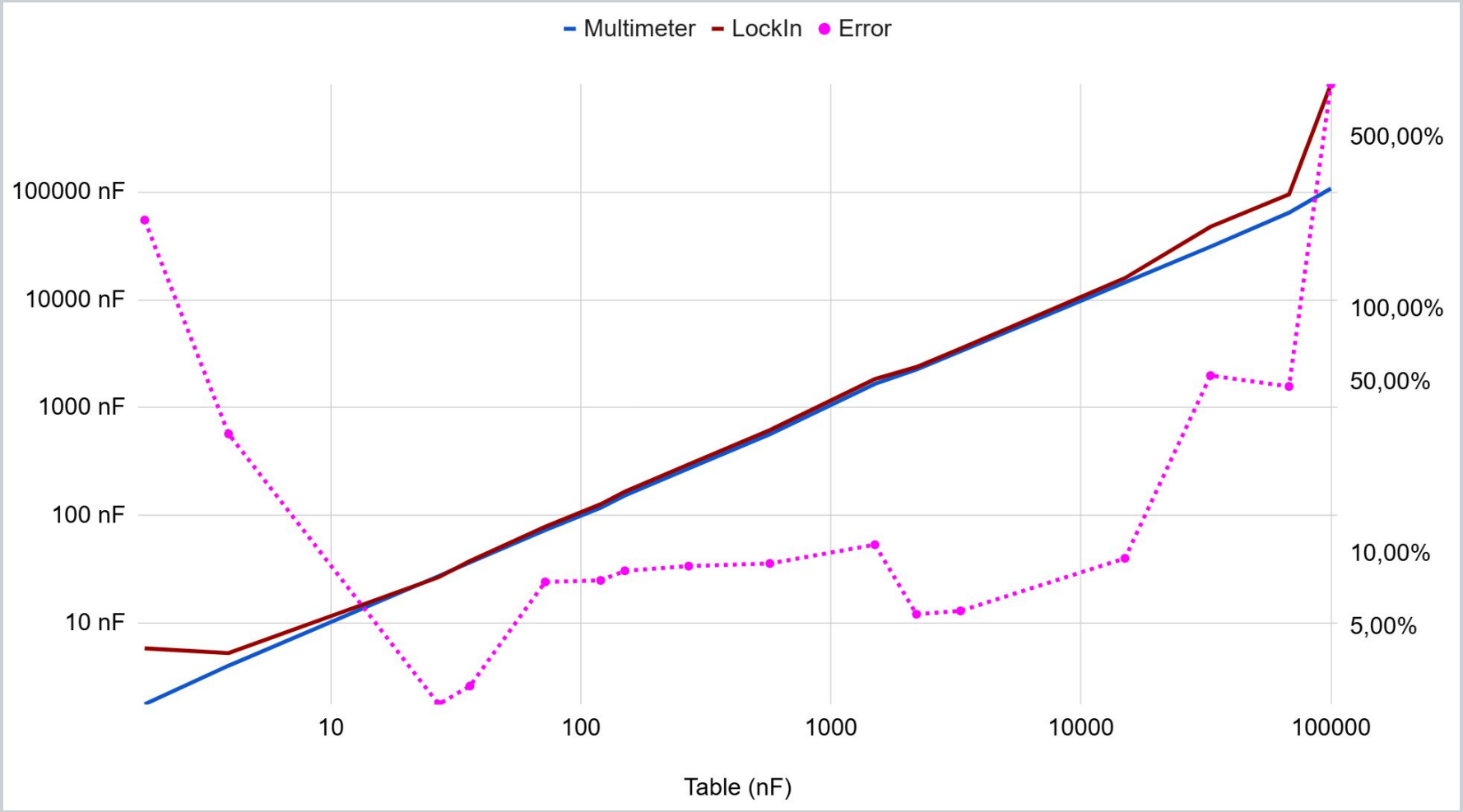
Por isso, foi decidido utilizar a linguagem C. Após realizar overclock do microcontrolador (para 270 MHz) e do ADC (para 135 MHz), obtemos uma medida a cada 0,71 µs.



Resultado em resistências



Resultado em capacitâncias



Pontos de melhoria

Reduzir o ruído de alimentação: desenvolver um módulo de alimentação capaz de gerar tensões de 5 V, 3.3 V e 1.65 V de forma estável e com baixo ruído.

Melhorar a resolução do ADC: o ADC da Raspberry Pi Pico W conta com vários problemas que contribuem para reduzir seu número efetivo de bits para 9 bits.

Variar a frequência da referência: manter o sinal de saída com a maior excursão possível é benéfico para o desempenho. Para isso, a frequência da entrada deve ser inversamente proporcional à capacitância a ser medida.

Obrigado pela atenção!

UNIVERSIDADE FEDERAL
DE SANTA CATARINA