



## Adaptação da Biblioteca Should-DSL para Testagem de Softwares Embarcados

*Thiago Campos Teixeira e Nascimento, Lucas Campos Teixeira e Nascimento, Rogério Atem de Carvalho*

Em sistemas embarcados, um ambiente de testagem efetivo é essencial para o desenvolvimento ágil de um projeto. Essa necessidade, porém, é prejudicada pela reduzida capacidade de memória e processamento inerentes a um microcontrolador, o que afeta diretamente a capacidade de transmitir rapidamente alterações ao sistema de forma dinâmica. No caso de processadores mais potentes, linguagens de programação interpretadas como Python se sobressaem ao permitir a execução de testes e modificações gerais no programa, de forma extremamente rápida. Sistemas embarcados, porém, geralmente não possuem a capacidade de executá-las devido as limitações de suas arquiteturas. Como forma de suprir essa demanda, foram criadas alternativas para trazer a tecnologia de Python para estes ambientes, sendo uma delas o interpretador MicroPython.

MicroPython disponibiliza a linguagem de programação Python 3 para sistemas embarcados de modo que seja possível enviar código e testá-lo de maneira rápida, além de disponibilizar um terminal interativo para envio de código imediato. Através dele, diversas bibliotecas estão presentes, porém algumas se encontram indisponíveis devido a limitação do ambiente embarcado, sendo uma delas a Should-DSL, uma API que expande as capacidades de Python de realizar testes, implementando diversas novas operações em um formato bem mais natural. A API possui uma quantidade grande de comportamentos que podem ser aplicados pelo programador, desde os mais simples aos mais complexos, removendo assim a necessidade de implementá-los manualmente na fase de testes. Devido as diferenças de comportamento e limitações de interpretadores, Should-DSL não é compatível com MicroPython e não pode ser utilizado diretamente. Então, para suprir esta necessidade, foi iniciado um projeto de conversão da biblioteca Should-DSL para MicroPython, que implementaria a maior quantidade de funções presentes na biblioteca original ao mesmo tempo que tenta se aproximar o máximo possível do comportamento esperado. Por conta das diferenças de arquitetura, foi necessário criar essa nova biblioteca do zero e suprimir algumas funcionalidades que não podem ser implementadas, no momento, pela ausência de funções análogas. Após estudar o interpretador e analisar as diferenças de comportamentos, a biblioteca está sendo desenvolvida com sucesso.

*Instituição do Programa de IC, IT ou PG: PIBITI  
Fomento da bolsa (quando aplicável): IFF*