

Introdução e Objetivos

Neste trabalho descrevemos os resultados preliminares de uma proposta de plataforma para automação de experimentos. A proposta é composta por uma interface de hardware composta por um microcontrolador PIC18F2550 e software de linguagem de programação, Python. Esta plataforma tem como objetivo estender as capacidades do computador pessoal em interagir com o ambiente externo, *i.e.* “sentir” o mundo físico através de sensores e através da interface proposta transferir a realidade física ao mundo virtual do computador. Optamos por utilizar um hardware de amplo acesso ao público em geral, evitando componentes SMD e outros elementos que possam dificultar a reprodução deste trabalho, reforçando desta forma na filosofia do hardware aberto.

Metodologia

Optou-se por uma interface de hardware utiliza um microcontrolador que permite uma conexão quase universal com dispositivos eletrônicos e sensores, possui entradas analógicas que permitem detectar sinais analógicos, por exemplo de sensores de temperatura analógico, entradas digitais, saídas digitais, saídas analógicas, protocolos de comunicação tipo SPI, I2C, entre outros. Como atualmente existe uma quase infinita opção de sensores comerciais, a interface proposta permite a conexão praticamente com qualquer uma delas, basta o usuário conhecer a forma como o respectivo sensor se comunica ou funciona. Atuadores podem ser acionados pelas saídas digitais e/ou analogias. Optou-se pelo PIC18F2550, pois proporciona montagem *through-hole* ou simplesmente *through-hole*, *i.e.* não é um componente de montagem de superfície (SMD) que traduz em uma enorme dificuldade de montagem a nível de protótipo ou a nível caseiro, a montagem SMD é muito utilizada na escala industrial por permitir uma redução significativa de custos de produção, contudo necessita de máquinas sofisticadas. Desta forma, a plataforma aqui proposta permite uma fácil reprodução.

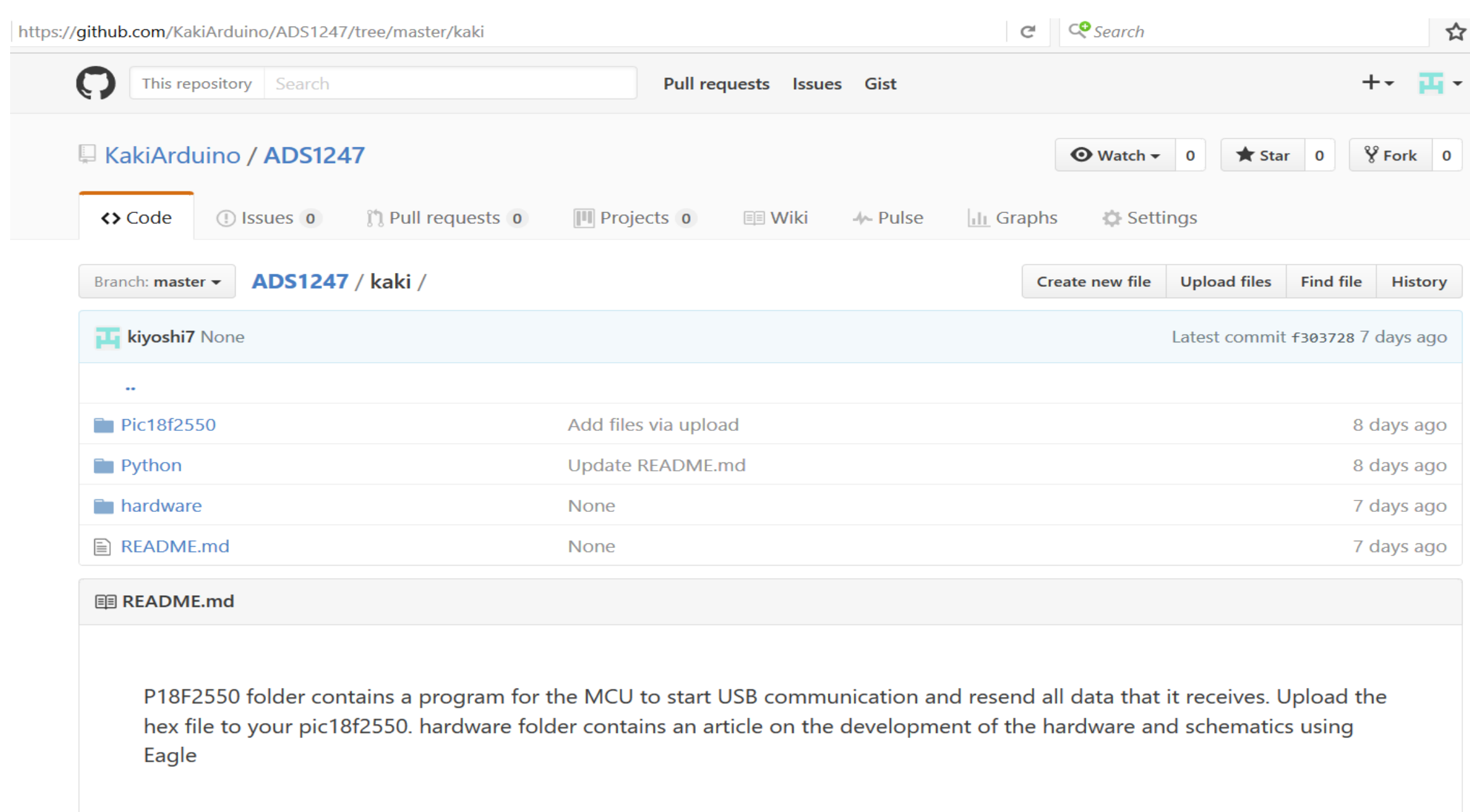


Fig1: Repositório no GITHUB.COM.

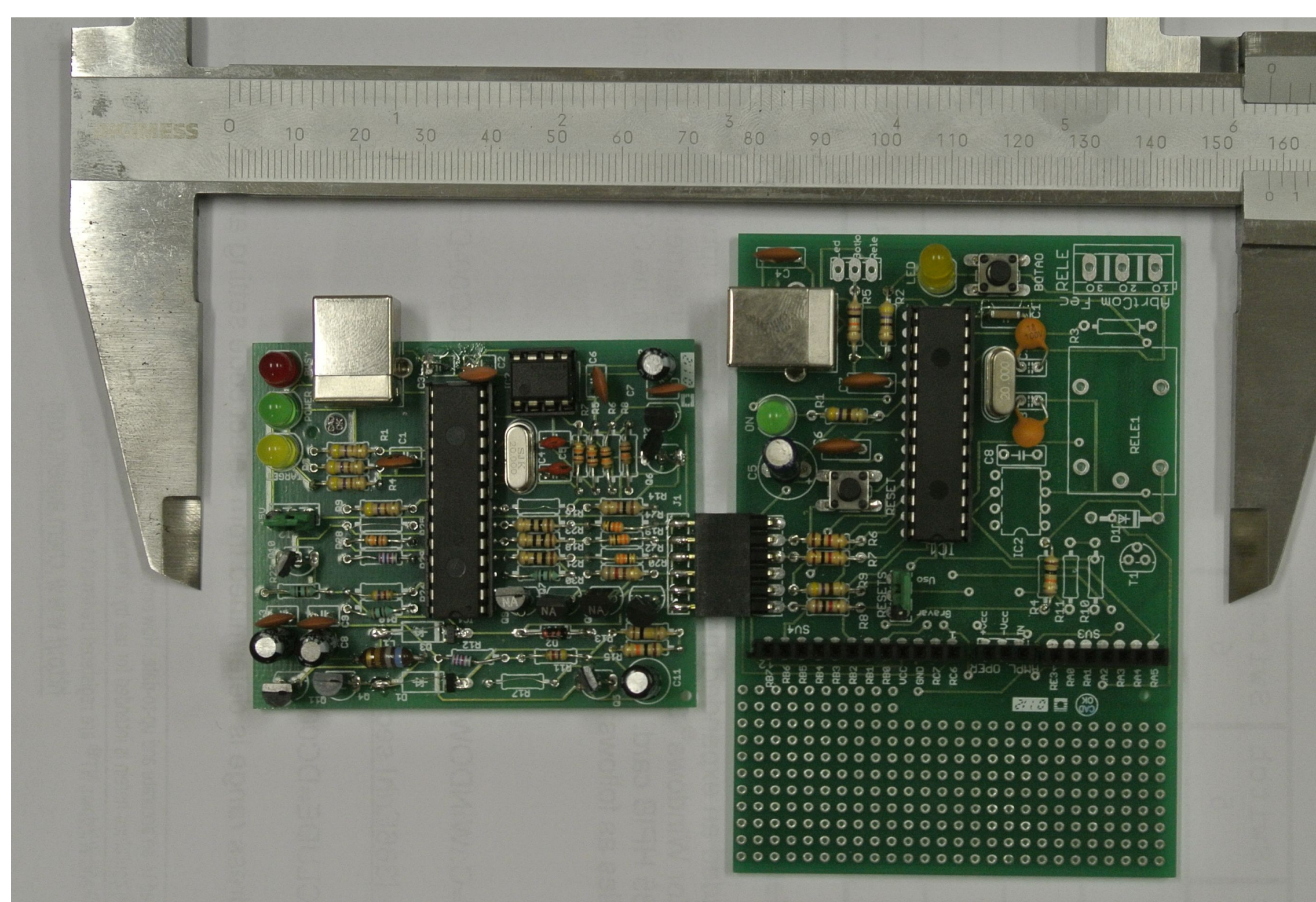


Fig2: Placa de desenvolvimento do PIC18F2550 à direita e placa gravador à esquerda.

Resultados e Discussões

Foi realizada comunicação com o computador através da interface USB (plenamente compatível com USB2.0) e protocolo HID, portanto não necessita a instalação de um driver específico. Optou-se pelo software Python, por ser um software não proprietário, de distribuição aberta, multiplataforma (*Macintosh, Windows e Linux*), muito poderoso e flexível, com extensa documentação. Neste projeto propomos desenvolver rotinas (para o Python) de leitura e escrita analógica, leitura e escrita digital e comunicação SPI em Python para que o usuário leigo não necessite programar o microcontrolador (PIC) em nível de dispositivo. Utilizamos um *software* proprietário (MikroC for PIC) para programa o PIC, *i.e.* desenvolver o *firmware* do microcontrolador, que iremos disponibilizar de forma livre.

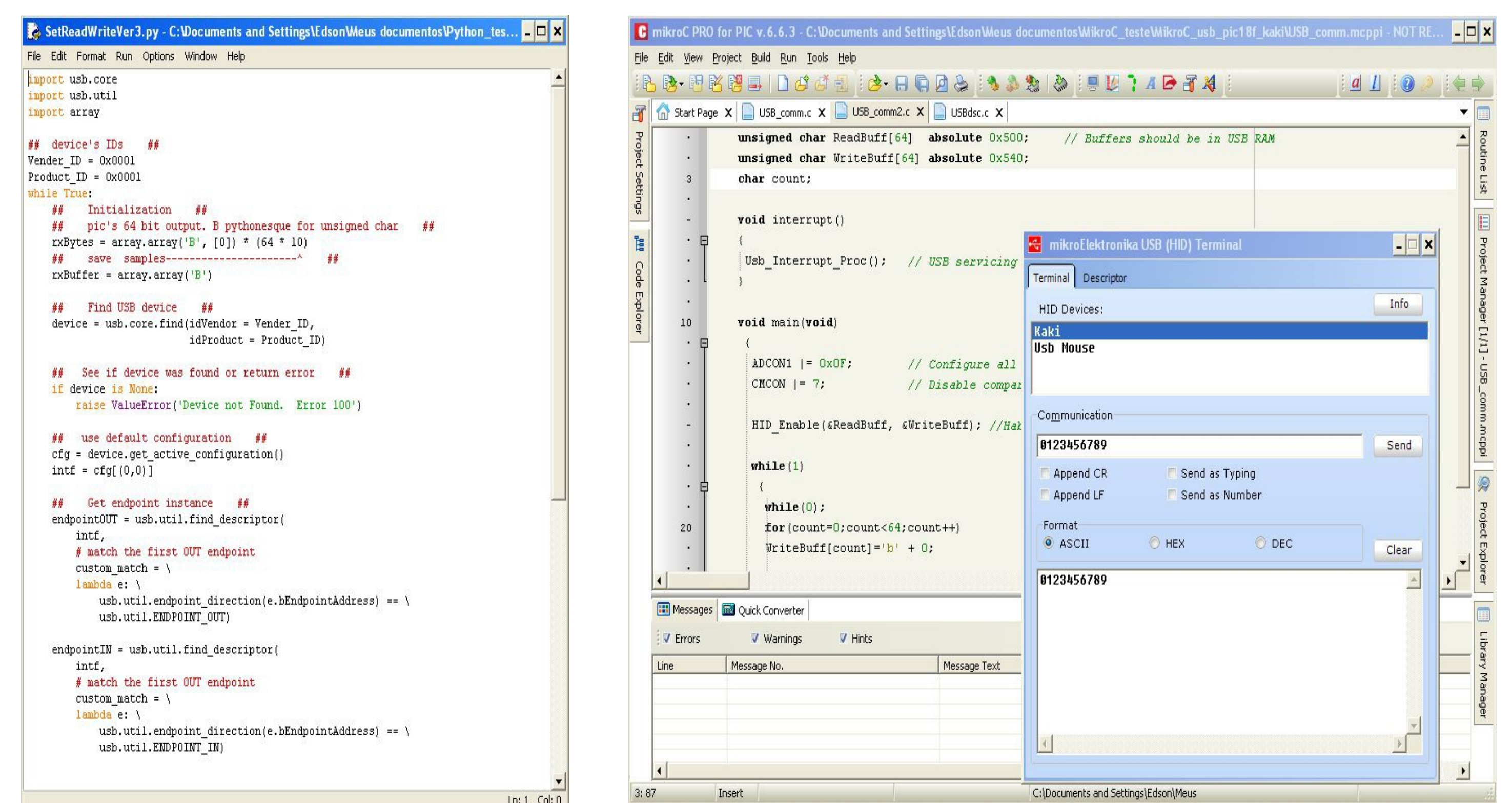


Fig3: Exemplo de código no Python, à esquerda e a direita exemplo de código no MikroC, com o terminal HID (comunicação via USB) aberto.

Conclusões

Atualmente temos uma plataforma básica de desenvolvimento do PIC18F2550 (placa de circuito impresso) e desenvolvemos com sucesso um gravador (clone do PicKit2 da Microchip, o qual o fabricante disponibiliza de forma pública) do PIC18F2550. Realizamos uma oficina de montagem do gravador e da plataforma de desenvolvimento e programação básica do PIC18F2550. A nível de software, estamos realizando comunicação básica entre o PIC18F2550 e o Python, através da porta USB, na plataforma Windows e Linux. Atualmente estamos desenvolvendo a comunicação SPI com os conversores analógicos digitais ADS1247 (Texas Instruments) e AD7794 (Analog Device), ambos com 24 bits de resolução. Este trabalho mantém cooperação com os Profs. Reginaldo Tavares (UFPEL) e Fabrício Ferrari (FURG).

Referências

- Microchip. Disponível em <<http://www.microchip.com/wwwproducts/en/PIC18F2550>> . Acessado em 20/10/2016.
- Microchip. Disponível em <<http://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf>> . Acessado em 20/10/2016.
- Python. Disponível em <<https://www.python.org/>> . Acessado em 20/10/2016.
- Python. Disponível em <<https://docs.python.org/3.5/using/mac.html>> . Acessado em 20/10/2016.
- KakiArduino. Disponível em <<https://github.com/KakiArduino/ADS1247/tree/master/kaki>> . Acessado em 20/10/2016.

Agradecimentos

O presente trabalho foi realizado com apoio do Programa Institucional de Bolsa de Iniciação à Docência – PIBID (Edital 2014), da CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil e impresso com apoio do Programa de Extensão Observatório de Aprendizagem (PROEXT-MEC 2009). A Unipampa pelo apoio ao projeto.