

# Co robimy?

Chcemy osiągnąć środowisko, w którym będzie można:

- pisać kod i go assemblować.
- debuggować kod (i widzieć które linie się wykonują w mnemonikach, a nie tylko w heksach).
- otoczyć nasz wirtualny mikroprocesor urządzeniami - diodami, oscyloskopami, przyciskami, wyświetlaczami i nauczyć go z nimi współpracować.

Chcielibyśmy, żeby uruchomienie tego środowiska wymagało jak najmniej pracy od potencjalnych użytkowników i żeby nasze środowisko było łatwe do rozszerzenia o dodatkowe urządzenia, które można podłączyć do mikroprocesora.

## Dlaczego to robimy?

Bo takie środowisko nie istnieje :-)

Istniejące otwarte emulatory:

- [gsim51](#) - służy tylko do uruchamiania już skompilowanych plików hex, wymaga kompilacji.
- [emu8051](#) - dokładnie jak wyżej, ponadto [wygląda paskudnie](#) .
- [emu51](#) - wymaga kompilacji, obecnie przystosowany tylko do windowsa, służy do uruchamiania plików hex, wymaga allegro w wersji z 2003 roku.

## Jak to robimy?

- Kod i błędy trzymamy na [Launchpadzie](#) .
- Stosujemy TDD, a testy piszemy w [JS-test-driver](#) .
- Stosujemy [następujące konwencje zapisu kodu](#) .

Wpisany przez Administrator  
sobota, 16 października 2010 19:17 -

---

Materiały dla członków zespołu:

- [Dobry tutorial pisania asma na 8051](#)
- [baza konkretnej wiedzy - opis rozkazów i opkody](#)
- [Dokumentacja JS-test-driver](#) - frameworku do testowania JS
- [Szybkie wprowadzenie do JS](#)
- [JSLint](#)
- [przykładowe code conventions](#) .