

Especificación de la API para el proyecto del BSVC

Ignacio Ballesteros González

22 de enero de 2018

La idea de esta guía es que se puedan implementar en cualquier lenguaje diferentes servidores *wrapper* para trabajar con el simulador *sim68000* del *BSVC*.

Métodos del servidor

La comunicación con el servidor se realiza mediante *WebSockets* con un continuo flujo de información entre los clientes y el servidor (o servidores).

Se podrán diferenciar distintos niveles de importancia según los métodos que el servidor implemente: **básicos** y **avanzados**. No se trata de un criterio de dificultad, sino de funcionalidad final.

Métodos básicos

Registros

Entrada en el simulador: `ListRegisters`

Respuesta: `[Str =>Hex]`

Comando (*WebSocket*) `registers`

Respuesta del server

```
{ "type": "array",  
  "items": {  
    "type": "object",  
    "properties": [  
      "name" : { "type": "string" },  
      "value": { "type": "string" }  
    ]  
  }  
}
```

Ejemplo de respuesta:

```
[ { "name": "D0", "value": "00000000" }  
  , { "name": "D1", "value": "00000000" }  
  , { "name": "D2", "value": "00000000" }  
  , { "name": "D3", ...
```

Memoria

Entrada en el simulador: `ListMemory AddrSpace InitialAddr Length WordsPerLine`

Respuesta: `[[Hex]]`

Comando (*WebSocket*) `memory initialAddr::Int [length::Int wpl::Int]`

Ejemplo
`memory 20`
`memory 20 10 9`

Respuesta del server

```
{ "type": "array",
  "items": {
    { "type": "array",
      "items": {
        "type": "string",
      }
    }
  }
}
```

El *Address Space* es siempre 0. Los valores de longitud y de palabras por línea son opcionales.

Sus valores por defecto son:

Longitud 4

Palabras por línea 16

Ejemplo de respuesta:

```
[ [ "c0", "de", "7f", "b3", "9d", ... ]
, [ "30", "30", "30", "30", "30", ... ]
, [ "e... ]
...
]
```