

Instrucciones para Agregar Nuevas Instrucciones AVR

Este proyecto utiliza definiciones de instrucciones en formato **JSON** (o Dart) para decodificar instrucciones AVR a partir de archivos `.hex`. Para agregar nuevas instrucciones, seguí estos pasos:

Estructura de una Instrucción

Cada instrucción está definida por un objeto con al menos los siguientes campos:

```
{  
  "bitPattern": "xxxx xxxx xxxx xxxx",  
  "nameTemplate": "INSTRUCCION {param1}, {param2}"  
}
```

Campos adicionales opcionales:

- "secondBitPattern" – si la instrucción ocupa 2 palabras (como CALL, JMP)
- "offsets" – si necesitás especificar de qué bit a qué bit extraer valores

Pasos para Agregar una Nueva Instrucción

1. Buscar en el manual AVR

Encontrá la instrucción y su codificación de opcodes, por ejemplo:

IN: 1011 0AA dddd AAAA

2. Escribir el bitPattern

Reemplazá los bits variables con letras que representen los campos:

```
"bitPattern": "1011 0AA dddd AAAA"
```

3. Escribir el nameTemplate

Usá las variables entre llaves {} para construir la representación legible. Ejemplo:

```
"nameTemplate": "IN R{d}, 0x{A}"
```

Podés usar operaciones dentro del nombre, como:

```
"nameTemplate": "MOVW R{2*d+1}:R{2*d}, R{2*r+1}:R{2*r}"
```

Reglas Especiales

- Instrucciones como BRxx, RCALL, RJMP usan un **offset relativo**.
 - Instrucciones con registros pares (R25:R24, MOVW, etc.) suelen requerir fórmulas como {2*d+24} para reflejar el mapeo real.
 - Algunas instrucciones (como CALL, JMP, LDS, STS) ocupan 32 bits → usá "secondBitPattern".
-

Ejemplo de Instrucciones

```
[
  {
    "bitPattern": "1001 11rd dddd rrrr",
    "nameTemplate": "MUL R{d}, R{r}"
  },
  {
    "bitPattern": "0110 KKKK dddd KKKK",
    "nameTemplate": "SBR R{d+16}, 0x{K}"
  },
  {
    "bitPattern": "1001 010k kkkk 111k",
    "secondBitPattern": "kkkk kkkk kkkk kkkk",
    "nameTemplate": "CALL 0x{k}"
  }
]
```

Dónde guardar las instrucciones

Agregá tus nuevas definiciones al archivo:

AVR_ISA.json

Plantilla rápida

```
{
  "bitPattern": "____ _",
  "nameTemplate": "INSTRUCCION {param1}, {param2}"
}
```