

AVR ADC Profile Switcher

This project provides a modular and flexible ADC configuration system for AVR microcontrollers (e.g., ATmega328P), enabling users to switch between predefined ADC profiles based on application needs.

Overview

The sketch demonstrates a hardware-triggered 8-bit ADC sampling approach using the internal timer and interrupt-driven data acquisition. It features a structured and extensible way of selecting between different ADC configurations, encapsulated in a single profile registry.

The code keeps `loop()` and `ISR()` lean for optimal performance and low power consumption, making it suitable for embedded applications where deterministic sampling and tight RAM usage are important.

Features

- Multiple ADC profiles stored in a centralized config structure
- Profile-based initialization via a simple `uint8_t` selector
- Timer1 Compare Match A as a hardware trigger source
- Optional free-running mode with internal 1.1V reference
- 8-bit sampling using left-adjusted ADC mode
- Circular buffer with fixed memory footprint (8 bytes)
- Ultra-lightweight ISR handling for fast ADC reads
- CPU sleep mode (`SLEEP_MODE_ADC`) to conserve power

Profiles Included

1. Timer1-triggered ADC (Profile 0)

- Triggered via Timer1 Compare Match A
- Reference: AVcc
- Left-adjusted 8-bit result (uses `ADCH`)
- Full ADC interrupt-driven sampling

2. Free-running ADC (Profile 1)

- Continuously samples as fast as possible
- Reference: Internal 1.1V
- Right-adjusted 10-bit result (still reads `ADCH` in current setup)
- Low-latency sampling without external triggers

Intended Use

This project is intended for developers looking to:

- Perform periodic ADC reads without blocking the CPU
- Toggle between precise timing and continuous free-running modes

- Experiment with low-power analog signal capture on AVRs
- Use a modular approach for embedded firmware configuration

Requirements

- Compatible with ATmega328P or similar AVR microcontrollers
- AVR-GCC or Arduino IDE with AVR toolchain
- (Optional) Serial monitor or logic analyzer for debugging

Folder Structure

```
📁 voltmeter/  
├─ src/ # Main sketch and logic  
├─ README.md # Project description and usage  
└─ LICENSE # (Optional) License file
```

License

See [LICENSE](#) for details.

Author

Hendrik Siemens — blending low-level control with modular, maintainable firmware design. 😊