

# FreeRTOS on Arduino Nano Every

The FreeRTOS contributor that has the Arduino port hasn't updated for the ATmega4809 in the Arduino Nano Every. I've managed to get it working. The mods do not apply to the Arduino Uno Wifi which also uses the ATmega4809. I did not put this in *Far Inside The Arduino: Nano Every Supplement*, so am posting what needs to be done here. You start with the current distribution from [https://github.com/feilipu/Arduino\\_FreeRTOS\\_Library](https://github.com/feilipu/Arduino_FreeRTOS_Library).

I've tried several sample programs and they all work, but this hasn't been tested thoroughly, so use at your own risk!

---

In `variants/nona4809/variant.c`, remove or comment out the `initVariant` function. This will not affect operation of any programs but allows FreeRTOS to compile.

The remaining changes are to the FreeRTOS library. The changes involve switching from using the WDT interrupt, which is not implemented in the ATmega4809, with the PIT interrupt. Interrupts will be roughly every 15ms, the same as the default WDT use in FreeRTOS.

The FreeRTOS type `TCB_t` conflicts with the ATmega4809 library. So change all “TCB\_t” to something else. I used “TCb\_t”.

In file `portable.h`, find the line `#include <avr/wdt.h>`. Add before that line:

```
#if defined (__AVR_ATmega4809__)  
#else
```

Go down to the line `#include “mpu_wrappers.h”` and add a line `#endif` before it.

In file `port.c`, find the line that defines `portScheduler_ISR` and replace it with:

```
#if defined (__AVR_ATmega4809__)  
#define portSCHEDULER_ISR RTC_PIT_vect  
#else  
#define portSCHEDULER_ISR WDT_vect
```

```
#endif
```

In the function vPortYieldFromTick after the line(s) portSAVE\_CONTEXT() add:

```
#if defined ( __AVR_ATmega4809__ )  
    RTC_PITINTFLAGS = 1; // clear interrupt  
#endif
```

Replace the function prvSetupTimerInterrupt with:

```
void prvSetupTimerInterrupt( void )  
{  
#if defined ( __AVR_ATmega4809__ )  
    RTC_CLKSEL = 0; // 32.767kHz  
    RTC_PITINTCTRL = 1; // Enable interrupt  
    RTC_PITCTRLA = ( 8 << 3 ) | 1; // Divide by 512; 15msec  
rate  
#else  
    /* reset watchdog */  
    wdt_reset();  
  
    /* set up WDT Interrupt (rather than the WDT Reset). */  
    wdt_interrupt_enable( portUSE_WDTO );  
#endif  
}
```

Before the ISR(portSCHEDULER\_ISR) functions, add:

```
#if defined ( __AVR_ATmega4809__ )  
__attribute(( naked, noline )) static void isrIrq0n () {  
    asm("reti");  
}  
#endif
```

In the first ISR(portSCHEDULER\_ISR) replace the single line that calls vPortYieldFromTick with:

```
#if defined ( __AVR_ATmega4809__ )  
    cli(); // Disable interrupts globally  
    isrIrq0n(); // do an RTI, leaving interrupts disabled  
#endif  
    vPortYieldFromTick();  
#if defined ( __AVR_ATmega4809__ )  
    sei(); // Reenable interrupts globally  
#endif
```

Finally, in the second ISR(portSCHEULER\_ISR), the one that calls xTaskIncrementTick, add the lines:

```
#if defined (__AVR_ATmega4809__)  
    RTC_PITINTFLAGS = 1; // clear interrupt  
#endif
```