

# Descripción general del oscilador megaAVR®

Los microcontroladores de 8 bits megaAVR® de Microchip tienen varias opciones de fuente de reloj, seleccionables a través de la programación de los bits de fusible **CKSEL Flash (/8avr:avrfuses)** . Esta discusión es específica de la MCU ATmega328PB . (<http://www.microchip.com/wwwproducts/en/ATmega328PB>) Los bits de fusible pueden seleccionar uno de:

- Oscilador de cristal de baja potencia
- Oscilador de cristal de baja frecuencia
- Oscilador RC interno de 128 kHz
- Oscilador RC interno calibrado, y
- Reloj externo.



La **fuente del reloj del sistema no se** puede cambiar durante el tiempo de ejecución, ya que se configura a través de la programación de fusibles.



La **frecuencia del reloj del sistema se** puede cambiar durante el tiempo de ejecución escribiendo en el registro del preescalador del **reloj del sistema (/8avr:osc-mega-overview#system-clock-prescaler)** (CLKPR).

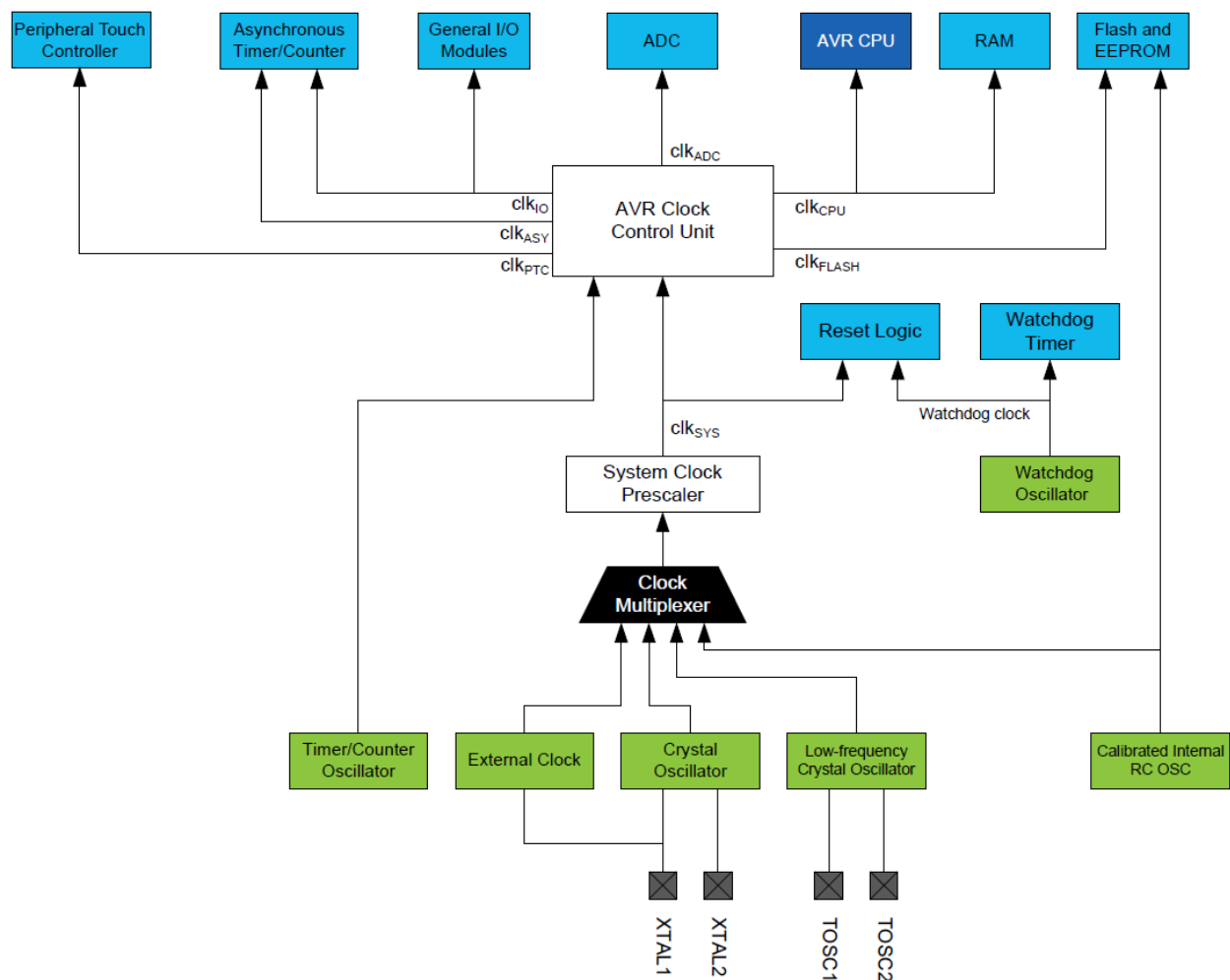
Cada fuente de reloj proporciona una opción de retraso después del reinicio o encendido del dispositivo para mantener el dispositivo reiniciado hasta que se suministre con Vcc mínimo. El reloj de la fuente seleccionada se ingresa al generador de reloj AVR® y se enruta a los módulos apropiados.



La **frecuencia operativa máxima del megaAVR® depende de V<sub>CC</sub>** . El software de la aplicación debe garantizar que la frecuencia de la fuente de reloj seleccionada se encuentre dentro del área de operación segura (consulte la sección 33.4 en la **hoja de datos del dispositivo** (<http://ww1.microchip.com/downloads/en/DeviceDoc/40001906A.pdf>) ).

## Visión general

La siguiente figura ilustra los principales sistemas de reloj del dispositivo y su distribución. No es necesario que todos los relojes estén activos en un momento dado. Para reducir el consumo de energía, los relojes de los módulos que no se utilizan se pueden detener utilizando diferentes modos de suspensión (/8avr:avrsleep) . Los sistemas de reloj se describen en las siguientes secciones. La frecuencia del reloj del sistema se refiere a la frecuencia generada por el preescalador del reloj del sistema. Todas las salidas de reloj de la unidad de control de reloj AVR funcionan a la misma frecuencia.



(/local--files/8avr:osc-mega-overview/atmega328pb-sys-clk-distribution.png)

## Fuentes de reloj

El dispositivo tiene las siguientes opciones de fuente de reloj, seleccionables a través de los bits **CKSEL** Flash Fuse como se muestra a continuación. El reloj de la fuente seleccionada se ingresa al generador de reloj AVR® y se enruta a los módulos apropiados.

Device Clocking Option	CKSEL[3:0]
Low Power Crystal Oscillator	1111 - 1000
Low Frequency Crystal Oscillator	0101 - 0100
Internal 128kHz RC Oscillator	0011
Calibrated Internal RC Oscillator	0010
External Clock	0000
Reserved	0001

(/local--files/8avr:osc-mega-overview/atmega328pb-clk-sources.png)

### Fuente de reloj predeterminada

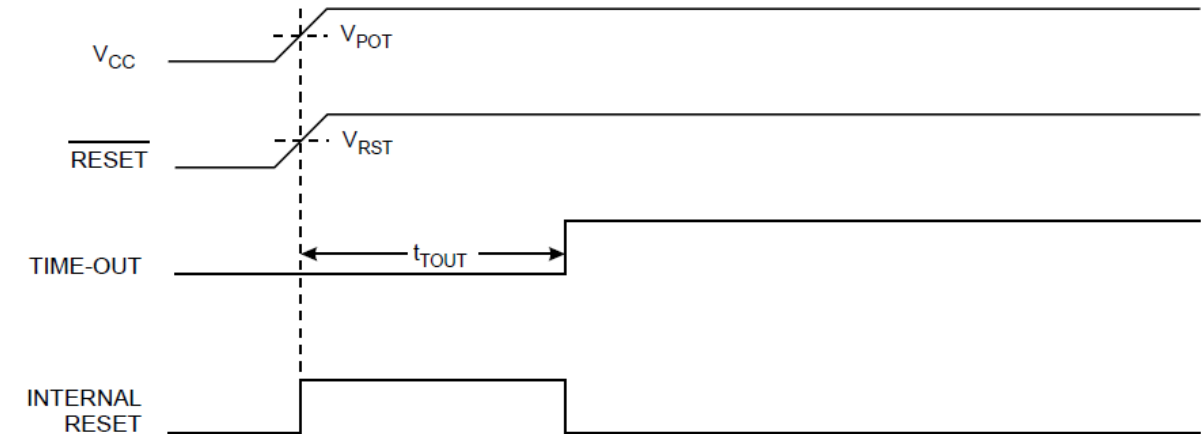
El dispositivo se envía con el oscilador RC interno seleccionado a 8,0 MHz y con el fusible CKDIV8 programado, lo que da como resultado un reloj del sistema de 1,0 MHz. El tiempo de inicio se establece al máximo y el período de tiempo de espera está habilitado: CKSEL=0010, SUT=10, CKDIV8=0. Esta configuración predeterminada garantiza que todos los usuarios puedan configurar la fuente de reloj deseada utilizando cualquier interfaz de programación disponible.

### Secuencia de inicio del reloj

Cualquier fuente de reloj necesita (i) **un  $V_{CC}$  suficiente para comenzar a oscilar** y (ii) **un número mínimo de ciclos de oscilación antes de que pueda considerarse estable** .

### Estabilidad $V_{CC}$

Para garantizar suficiente  $V_{CC}$  , el dispositivo emite un restablecimiento interno con un retraso de tiempo de espera (  $t_{TOUT}$  ) después de que todas las demás fuentes de restablecimiento liberan el restablecimiento del dispositivo:



(/local--files/8avr:osc-mega-overview/atmega328pb-tout-delay.png)

El retardo (  $t_{TOUT}$  ) se cronometra desde el oscilador de vigilancia y el tiempo de retardo se establece mediante los bits de fusible SUTx y CKSELx. Los retardos seleccionables para  $t_{TOUT}$  se muestran en la siguiente tabla. Tenga en cuenta que la frecuencia del Watchdog Oscillator depende del voltaje:

Typ. Time-out ( $V_{CC} = 5.0V$ )	Typ. Time-out ( $V_{CC} = 3.0V$ )
0ms	0ms
4ms	4.3ms
65ms	69ms

(/local--files/8avr:osc-mega-overview/atmega328pb-tout-values.png)



$V_{CC}$  no se controla durante el retraso, por lo que se requiere seleccionar un retraso más largo que el tiempo de subida de  $V_{CC}$ . Si esto no es posible, se debe utilizar un circuito de detección de Brown-Out (BOD) interno o externo. Un circuito BOD garantizará suficiente  $V_{CC}$  antes de liberar el reinicio, y el retardo de tiempo de espera se puede desactivar. No se recomienda deshabilitar el retardo de tiempo de espera sin utilizar un circuito de detección de Brown-Out.

## Estabilidad del oscilador

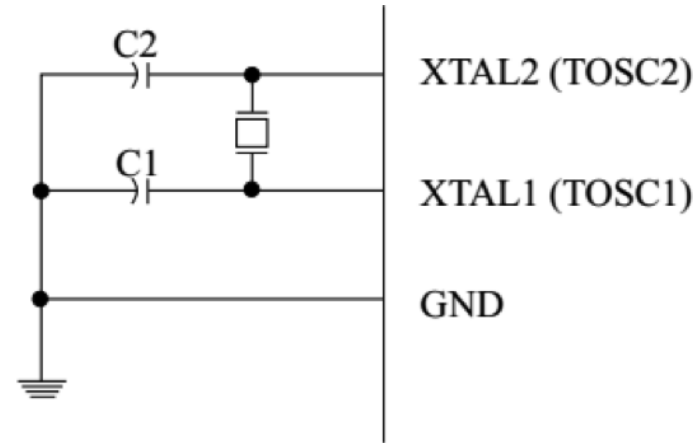
Se requiere que el oscilador oscile durante un número mínimo de ciclos antes de que el reloj se considere estable. Un contador de ondulación interno monitorea el reloj de salida del oscilador y mantiene activo el restablecimiento interno durante un número determinado de ciclos de reloj. Luego se libera el reinicio y el dispositivo comenzará a ejecutarse. El tiempo recomendado de puesta en marcha del oscilador depende del tipo de reloj y varía de 6 ciclos para un reloj aplicado externamente a 32K ciclos para un cristal de baja frecuencia.



Consulte la sección 11 en la hoja de datos del dispositivo (<http://ww1.microchip.com/downloads/en/DeviceDoc/40001906A.pdf>) que especifica el número de ciclos de retardo de CK para cada tipo de fuente de reloj y ajuste de fusible SUTx.

## Oscilador de cristal de baja potencia

Los pines XTAL1 y XTAL2 son entrada y salida, respectivamente, de un amplificador inversor que puede configurarse para usarse como un oscilador en chip, como se muestra en la figura a continuación. Puede usarse un cristal de cuarzo o un resonador cerámico:



(/local--files/8avr:osc-mega-overview/atmega328pb-xtal-connection.png)

El oscilador de baja potencia puede operar en tres modos diferentes, cada uno optimizado para un rango de frecuencia específico. El modo de operación es seleccionado por los fusibles CKSEL[3:1], como se muestra en la siguiente tabla:

Frequency Range [MHz]	CKSEL[3:1] <sup>(2)</sup>	Range for total capacitance of C1 and C2 [pF] <sup>(4)</sup>
0.4 - 0.9	100 <sup>(3)</sup>	–
0.9 - 3.0	101	12 - 22
3.0 - 8.0	110	12 - 22
8.0 - 16.0	111	12 - 22

(/local--files/8avr:osc-mega-overview/atmega328pb-xtal-modes.png)

El Fusible CKSEL0 junto con los Fusibles SUT[1:0] seleccionan los tiempos de arranque (ver sección 11.3 en la **hoja de datos del dispositivo** (<http://ww1.microchip.com/downloads/en/DeviceDoc/40001906A.pdf>) ).

### Oscilador de cristal de baja frecuencia

El oscilador de cristal de baja frecuencia está optimizado para su uso con un cristal de reloj de 32,768 kHz. El oscilador de cristal de baja frecuencia debe seleccionarse configurando los fusibles CKSEL en '0110' o '0111', y los tiempos de inicio están determinados por el SUT fusibles

### Oscilador RC interno calibrado

De forma predeterminada, el oscilador RC interno proporciona un reloj de 8,0 MHz. Aunque depende del voltaje y la temperatura, el usuario puede calibrar este reloj con mucha precisión. El dispositivo se envía con el fusible CKDIV8 programado, lo que proporciona una frecuencia de reloj del sistema de 1 MHz. Este reloj se puede seleccionar como el reloj del sistema programando los fusibles CKSEL en '0010':. Si se

selecciona, funcionará sin componentes externos. Durante el reinicio, el hardware carga el valor de calibración preprogramado en el registro OSCCAL y, por lo tanto, calibra automáticamente el oscilador RC.



Consulte la **nota de aplicación AVR053** ([http://ww1.microchip.com/downloads/en/AppNotes/Atmel-2555-Internal-RC-Oscillator-Calibration-for-tinyAVR-and-megaAVR-Devices\\_ApplicationNote\\_AVR053.pdf](http://ww1.microchip.com/downloads/en/AppNotes/Atmel-2555-Internal-RC-Oscillator-Calibration-for-tinyAVR-and-megaAVR-Devices_ApplicationNote_AVR053.pdf)) que describe el procedimiento para volver a calibrar el oscilador RC interno.

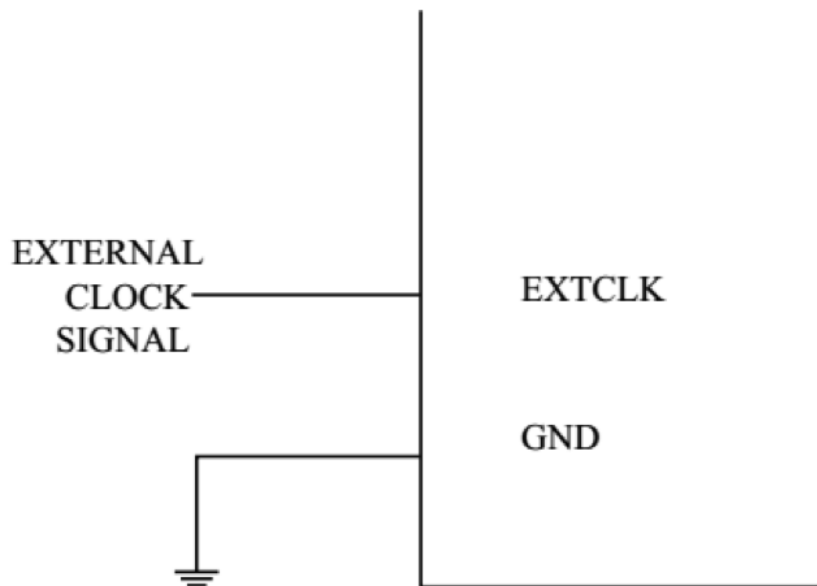
## Oscilador interno de 128 kHz

El oscilador interno de 128 kHz es un oscilador de baja potencia que proporciona un reloj de 128 kHz. Este reloj se puede seleccionar como reloj del sistema programando los fusibles CKSEL en '0011'.

## Reloj externo

Para controlar el dispositivo desde una fuente de reloj externa, EXTCLK debe controlarse como se muestra en la figura a continuación. Para ejecutar el dispositivo en un reloj externo, los fusibles CKSEL deben programarse en '0000'.

### External Clock Drive Configuration



(/local--files/8avr:osc-mega-overview/atmega328pb-ext-clk-connection.png)

## Búfer de salida de reloj

El dispositivo puede generar el reloj del sistema en el pin CLKO. Para habilitar la salida, se debe programar el Fusible CKOUT. Este modo es adecuado cuando el reloj del chip

se usa para controlar otros circuitos en el sistema. El reloj también se emitirá durante el reinicio, y el funcionamiento normal del pin de E/S se anulará cuando se programe el fusible. Cualquier fuente de reloj, incluido el oscilador RC interno, se puede seleccionar cuando el reloj se emite en CLKO. Si se utiliza el preescalador de reloj del sistema, lo que se emite es el reloj del sistema dividido.

## Temporizador/Contador Oscilador

El dispositivo utiliza el mismo oscilador de cristal para el oscilador de baja frecuencia y el temporizador/contador de osciladores. Consulte Oscilador de cristal de baja frecuencia para obtener detalles sobre los requisitos del oscilador y del cristal.

En este dispositivo, los pines del oscilador del temporizador/contador (TOSC1 y TOSC2) se comparten con EXTCLK. Cuando se utiliza el temporizador/contador de osciladores, el reloj del sistema debe ser cuatro veces la frecuencia del oscilador. Debido a esto y al uso compartido de pines, el temporizador/contador de osciladores solo se puede usar cuando el oscilador RC interno calibrado se selecciona como fuente de reloj del sistema. Se puede aplicar una fuente de reloj externa a TOSC1 si el bit Habilitar entrada de reloj externo en el registro de estado asíncrono (ASSR.EXCLK) se escribe en '1'. Consulte la descripción de la operación asíncrona del temporizador/contador2 para obtener una descripción más detallada sobre la selección de un reloj externo como entrada en lugar de un cristal de reloj de 32,768 kHz.

## Prescaler del reloj del sistema

El dispositivo tiene un preescalador de reloj del sistema, y el reloj del sistema se puede dividir configurando el Registro de preescala de reloj (CLKPR). Esta función se puede utilizar para disminuir la frecuencia del reloj del sistema y el consumo de energía cuando el requisito de potencia de procesamiento es bajo. Esto se puede usar con todas las opciones de fuente de reloj y afectará la frecuencia de reloj de la CPU y todos los periféricos síncronos.  $\text{clk}_{\text{I/O}}$ ,  $\text{clk}_{\text{ADC}}$ ,  $\text{clk}_{\text{CPU}}$  y  $\text{clk}_{\text{FLASH}}$  se dividen por un factor como se muestra en la descripción de CLKPR:

**Name:** CLKPR  
**Offset:** 0x61  
**Reset:** Refer to the bit description  
**Property:** -

Bit	7	6	5	4	3	2	1	0
	CLKPCE				CLKPSn	CLKPSn	CLKPSn	CLKPSn
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				x	x	x	x

#### Bit 7 – CLKPCE: Clock Prescaler Change Enable

The CLKPCE bit must be written to logic one to enable change of the CLKPS bits. The CLKPCE bit is only updated when the other bits in CLKPR are simultaneously written to zero. CLKPCE is cleared by hardware four cycles after it is written or when CLKPS bits are written. Rewriting the CLKPCE bit within this time-out period does neither extend the time-out period, nor clear the CLKPCE bit.

#### Bits 3:0 – CLKPSn: Clock Prescaler Select n [n = 3:0]

These bits define the division factor between the selected clock source and the internal system clock. These bits can be written run-time to vary the clock frequency to suit the application requirements. As the divider divides the master clock input to the MCU, the speed of all synchronous peripherals is reduced when a division factor is used. The division factors are given in the table below.

(/local--files/8avr:osc-mega-overview/atmega328pb-clkpr-1.png)

CLKPS[3:0]	Clock Division Factor
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001	Reserved
1010	Reserved
1011	Reserved
1100	Reserved
1101	Reserved
1110	Reserved
1111	Reserved

(/local--files/8avr:osc-mega-overview/atmega328pb-clkpr-2.png)

## Escribiendo a CLKPR

Al cambiar entre las configuraciones del preescalador, el Preescalador del reloj del sistema asegura que no ocurran fallas en el sistema del reloj. También asegura que ninguna frecuencia intermedia sea superior a la frecuencia de reloj correspondiente a la



configuración anterior, ni a la frecuencia de reloj correspondiente a la nueva

configuración. El contador de ondas que implementa el preescalador se ejecuta a la frecuencia del reloj indiviso, que puede ser más rápido que la frecuencia del reloj de la CPU. Por lo tanto, no es posible determinar el estado del preescalador; incluso si fuera legible, el tiempo exacto que se tarda en cambiar de una división de reloj a otra no se puede predecir con exactitud. Desde el momento en que se escriben los valores de los bits de selección del preescalador de reloj (CLKPS[3:0]), transcurren entre  $T1 + T2$  y  $T1 + 2 * T2$  antes de que se active la nueva frecuencia de reloj. En este intervalo, se producen dos flancos de reloj activos. Aquí,  $T1$  es el período de reloj anterior y  $T2$  es el período correspondiente a la nueva configuración del preescalador. Para evitar cambios involuntarios de la frecuencia del reloj, se debe seguir un procedimiento de escritura especial para cambiar los bits CLKPS:

1. Escriba el bit de habilitación de cambio de preescalador de reloj (CLKPCE) en '1' y todos los demás bits en CLKPR en cero: CLKPR=0x80.
2. Dentro de cuatro ciclos, escriba el valor deseado en CLKPS[3:0] mientras escribe un cero en CLKPCE: CLKPR=0x0N



Las interrupciones deben desactivarse al cambiar la configuración del preescalador para asegurarse de que el procedimiento de escritura no se interrumpa.

## Ejemplo de código

La siguiente función se puede utilizar para actualizar dinámicamente CLKPR como se requiere anteriormente. Tenga en cuenta el uso de las funciones `cli()` y `sei()` para garantizar que el procedimiento de escritura CLKPR no se interrumpa.



```
1  #include <stdint.h> // St?
2  #include <avr/io.h> // SFR
3  #include <avr/interrupt.h> /
4
5  void clkPrescaleSet(uint8_t
6      cli();
7      CLKPR = (1 << CLKPCE);
8      CLKPR = divisionFactor;
9      sei();
10 }
```



Para ver esta función en uso, visite el proyecto de ejemplo del **oscilador megaAVR®** (/8avr:osc-mega-example)

## CLKDIV8 Fusible y CLKPR

El fusible CKDIV8 determina el valor inicial de los bits CLKPS. Si CKDIV8 no está programado, los bits CLKPS se restablecerán a "0000". Si se programa CKDIV8, los

programado, los bits CLKPS se restablecerán a "0000". Si se programa CKDIV8, los bits CLKPS se restablecen a "0011", dando un factor de división de 8 al inicio. Esta función debe utilizarse si la fuente de reloj seleccionada tiene una frecuencia superior a la frecuencia máxima del dispositivo en las condiciones de funcionamiento actuales. Tenga en cuenta que se puede escribir cualquier valor en los bits CLKPS independientemente de la configuración del fusible CKDIV8. El software de la aplicación debe garantizar que se elija un factor de división suficiente si la fuente de reloj seleccionada tiene una frecuencia superior a la frecuencia máxima del dispositivo en las condiciones de funcionamiento actuales. El dispositivo se envía con el fusible CKDIV8 programado.

## Aprende más



### Proyecto de ejemplo de oscilador megaAVR®

Más información > (/8avr:osc-mega-example)