

Proyecto de ejemplo de fuentes de restablecimiento de AVR®

🕒 Objetivo

Este proyecto pasa por varias condiciones de reinicio diferentes: reinicio *de encendido (POR)* , reinicio *de apagado (BOR)* y tiempo de espera *del temporizador de vigilancia (WDT)* , y muestra cómo funciona cada uno en una **placa Xplained** de 328 PB . Se muestra que algunos circuitos externos producen una entrada de voltaje variable, pero también funcionará una fuente de alimentación ajustable.

Para obtener más detalles sobre las **fuentes de restablecimiento de AVR®**, visite Descripción general de las fuentes de restablecimiento de AVR. (/8avr:avrreset)

☑ Materiales

Herramientas de hardware (*opcional*)

Herramienta	📖 Sobre
 http://www.atmel.com/tools/MEGA328PB-XMINI.aspx	Mini kit de evaluación ATmega328PB Xplained ? (http://www.atmel.com/tools/MEGA328PB-XMINI.aspx)
	🛒 (https://www.microchipdirect.com)

Herramientas de software

Instaladores					
Herramienta	📖 Sobre	Windows	linux	Mac OS X	Instrucciones de instalación
Entorno de desarrollo integrado Atmel® Studio	📄 (/atstudio:start)	⬇️ (http://studio.download.atmel.com/7.0.1931/as-installer-7.0.1931-full.exe)	⬇️	⬇️	📄 (/install:atstudi

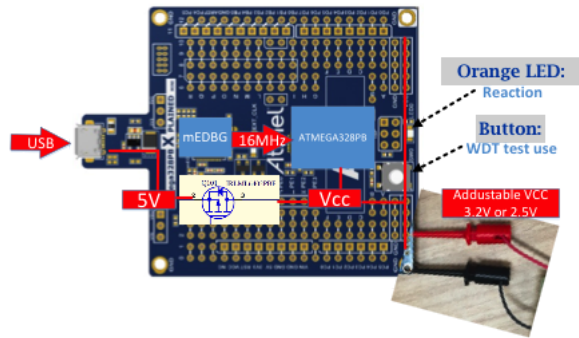
Archivos de ejercicios

Expediente	Windows
Archivos de proyecto y fuente	⬇️ (https://microchiptechnology.sharepoint.com/:u:/s/DeveloperHelp/ER6vOhSPHdIGtX1waUOOSH4BJ-IfY1jL6aKVwgc2Q8MizA?e=3PW9UP)

Archivos adicionales

archivos
 Guía del usuario de la miniplaca Xplained 328PB (http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42469-ATmega328PB-Xplained-Mini_User-Guide.pdf)

🔌 Diagrama de conexión

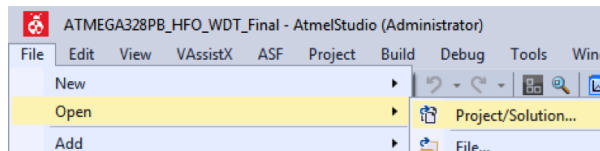


(/local--files/8avr:resetexample/project1.png)

Procedimiento

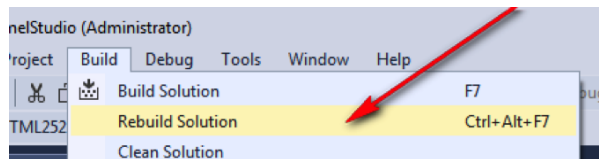
1 Tarea 1: abrir y compilar el proyecto

- Descargue los archivos fuente del proyecto de la sección anterior Archivos de ejercicios y descompríalos en su computadora.
- Abrir Atmel Studio 7
- Seleccione **Archivo > Abrir > Proyecto/Solución**



(/local--files/8avr:resetexample/step1.png)

- Seleccione `ATMEGA328PB_HF0_WDT_Final.atsln` de los archivos de proyecto descargados.
- En el menú Generar, seleccione 'Reconstruir solución'.

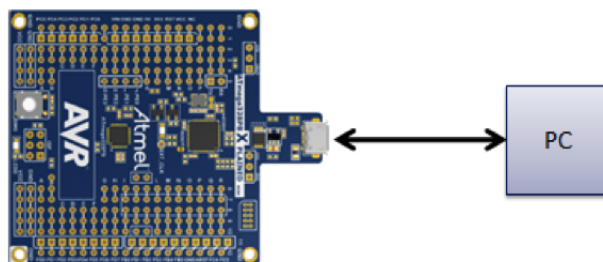


(/local--files/8avr:resetexample/step2.png)

Seleccione 'GCC C Executable Project' y asígnele el nombre `Project1`. Elija una ubicación para guardar el proyecto en su computadora.

2 Tarea 2: preparación de la placa

Asegúrese de que el cable USB esté conectado entre la placa y la PC.



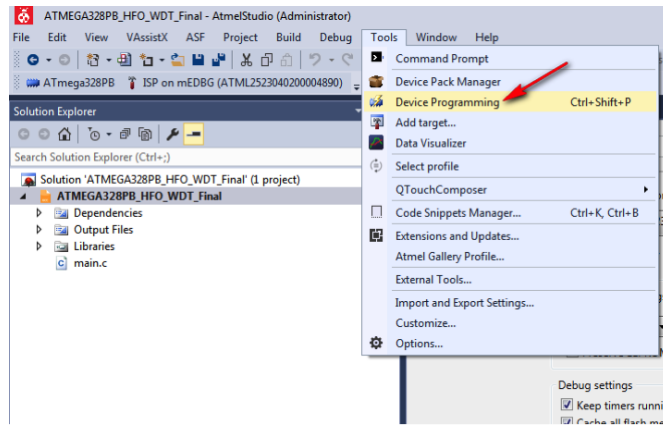
(/local--files/8avr:resetexample/step3.png)

3 Tarea 3 - Configuración del programador

Si debugWire está habilitado, desactívelo.

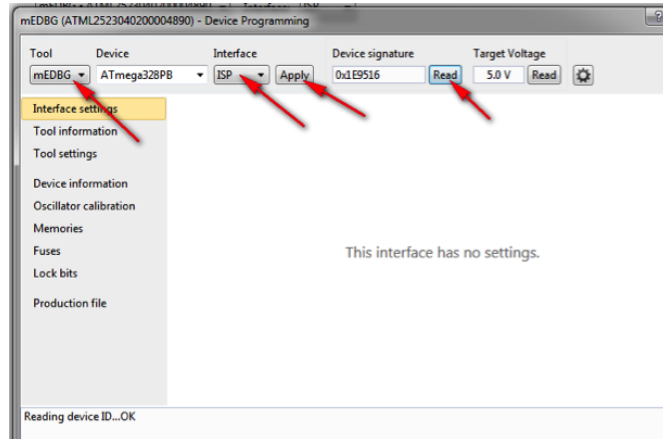
Deshabilitar debugWire (DWEN) ►

- Haga clic en **Herramientas > Programación de dispositivos** :



(/local--files/8avr:resetexample/step4.png)

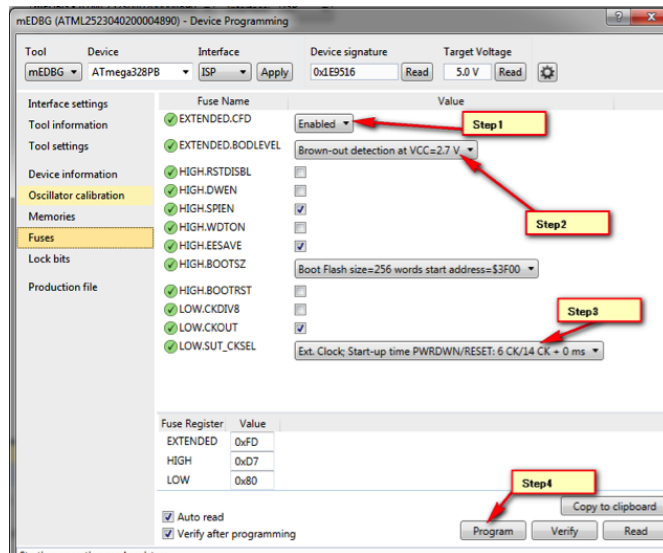
- Haga clic en los botones que están marcados como flechas rojas en la captura de pantalla.



(/local--files/8avr:resetexample/step5.png)

4 Tarea 4 - Programación de los bits de fusible

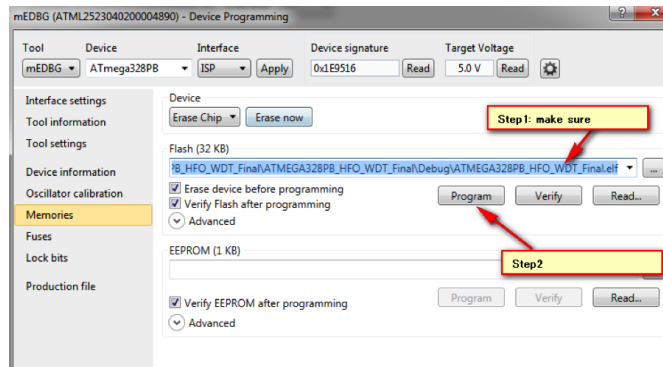
- Seleccione las opciones de 'Fusibles' en la barra de opciones de la izquierda y establezca los bits como se muestra en la figura, lo que habilitará CFD, umbral de DBO como 2,7 V y reloj externo.



(/local--files/8avr:resetexample/step6.png)

5 Tarea 5 - Programación de dispositivos

Seleccione las opciones de 'Memorias' en la barra de opciones de la izquierda y termine los pasos como se muestra en la figura, que programa el archivo binario compilado en ATMEGA328PB.



(/local--files/8avr:resetexample/step7.png)

6 Tarea 6 - Código del proyecto

Aquí está el código completo al que hacemos referencia.

También puedes descargarlo en la Sección de Ejercicios.

```

/*
 * ATMEGA328PB_HF0_WDT_Final.c
 *
 * Created: 2017/03/08 17:15:35
 * Author : A17582
 */

#define F_CPU 8000000UL

#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/wdt.h>
#include <util/delay.h>

//initialize watchdog
void WDT_Init(void)
{
    //disable interrupts
    cli();
    WDTCSR = (1<<WDCE)|(1<<WDE); // Enable configuration change.
    WDTCSR = (1<<WDIE)| // Enable Watchdog Interrupt Mode.
    (1<<WDCE)|(1<<WDE); // Enable Watchdog System Reset Mode if unintentionally enabled.
    (0<<WDP3)|(1<<WDP2)|(1<<WDP1)|(1<<WDP0); // Set Watchdog Timeout period to 4.0 sec.
    //Enable global interrupts
    sei();
}

//Watchdog timeout ISR
ISR(WDT_vect)
{
    //Burst of 0.1Hz pulses
    for (uint8_t i=0;i<4;i++)
    {
        //LED OFF
        PORTB &= ~(1 << PINB5); // Set PORTB5 Low
        _delay_ms(80);
        //LED ON
        PORTB |= (1 << PINB5); // Set PORTB5 On
        _delay_ms(20);
    }
}

#define B0RFbit 2
#define P0RFbit 0

int main(void)
{
    unsigned char i;
    DDRB |= (1 << PINB5); // Set PORTB5 as output ,
    DDRB &= ~(1<<PINB7); //Set PORTB7 as input

    if(MCUSR & 1){
        MCUSR=0;
        for ( i=0;i<4;i++)
        {
            //LED OFF
            PORTB &= ~(1 << PINB5); // Set PORTB5 Low
            _delay_ms(300);
            //LED ON
            PORTB |= (1 << PINB5); // Set PORTB5 On
            _delay_ms(300);
        }
    }
    else if(MCUSR & 4) {
        MCUSR=0;
        for (i=0;i<8;i++)
        {
            //LED OFF
            PORTB &= ~(1 << PINB5); // Set PORTB5 Low
            _delay_ms(100);
            //LED ON
            PORTB |= (1 << PINB5); // Set PORTB5 On
        }
    }
}

```

```

        _delay_ms(100);
    }
}
else if(MCUSR & 8) {
    MCUSR=0;
    WDT_Init();
    for (i=0;i<8;i++)
    {
        wdt_reset();
        //LED OFF
        PORTB &= ~(1 << PINB5);          // Set PORTB5 Low
        _delay_ms(20);
        //LED ON
        PORTB |= (1 << PINB5);            // Set PORTB5 On
        _delay_ms(80);
    }
    MCUSR=0;
}

//initialize watchdog
WDT_Init();

//delay to detect reset
//_delay_ms(500);

while(1){

    PORTB |= (1 << PINB5);                // Set PORTB5 high
    _delay_ms(250);
    _delay_ms(250);
    _delay_ms(250);
    _delay_ms(250);

    PORTB &= ~(1 << PINB5);                // Set PORTB5 Low
    _delay_ms(250);
    _delay_ms(250);
    _delay_ms(250);
    _delay_ms(250);

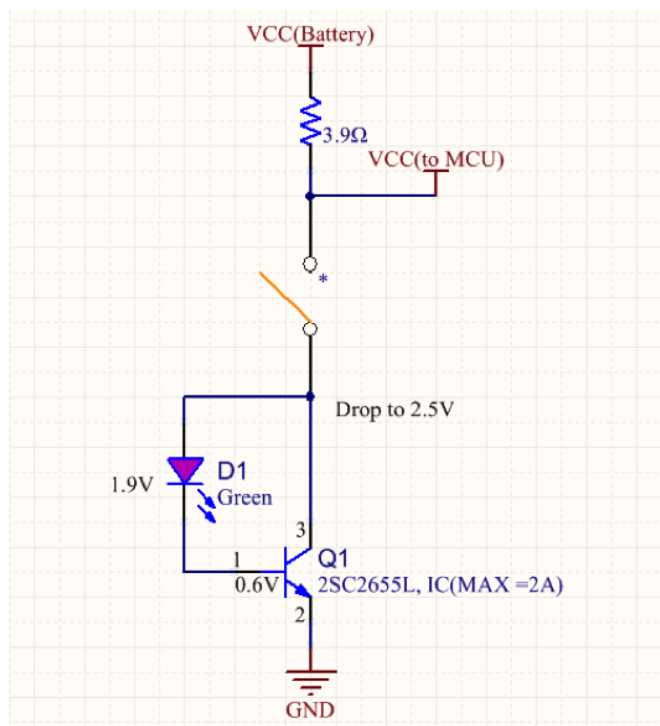
    if((PINB&1<<PINB7)==0){
        PORTB |= (1 << PINB5);            // Set PORTB5 high
        _delay_ms(500);
        _delay_ms(500);
        _delay_ms(500);
        _delay_ms(500);
        _delay_ms(500);
        _delay_ms(500);
        _delay_ms(500);
        _delay_ms(500);

    }
    wdt_reset();
}
}

```

7 Tarea 7: reinicio del circuito de prueba

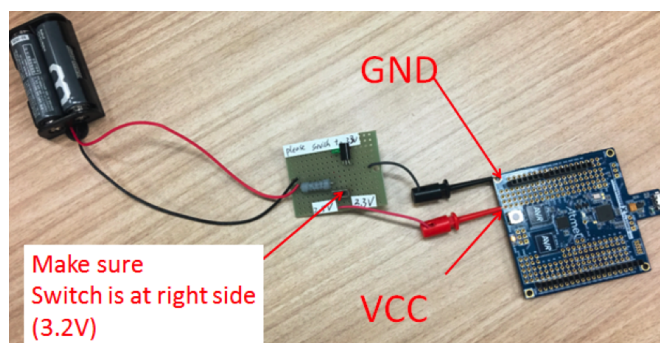
- Construya el circuito de prueba de reinicio que se muestra en el esquema.



(/local--files/8avr:resetexample/step12.png)

7 Tarea 8 - Prueba POR

- Saque el cable USB
- Asegúrese de que el interruptor del cable VCC ajustable externo esté en el lado derecho (3,2 V)
- Sujete el cable VCC ajustable externo a la placa



(/local--files/8avr:resetexample/step9.png)

- Resultado: el LED naranja responderá parpadeando (0,3 ms, cuatro veces) según la sección de código a continuación porque se detectó el restablecimiento de POR.

```
int main(void)
{
    unsigned char i;
    DDRB |= (1 << PINB5);           // Set PORTB5 as output
    DDRB &= ~(1<<PINB7);           //Set PORTB7 as input

    if(MCUSR & 1){                  //POR detected
        MCUSR=0;
        for ( i=0;i<4;i++)
        {
            //LED OFF
            PORTB &= ~(1 << PINB5); // Set PORTB5 Low
            _delay_ms(300);
            //LED ON
            PORTB |= (1 << PINB5);   // Set PORTB5 On
            _delay_ms(300);
        }
    }
    else if(MCUSR & 4) {
        MCUSR=0;
    }
}
```

(//local--files/8avr:resetexample/step10.png)

9 Tarea 9 - Prueba BOR

- Después de la prueba POR, deslice el interruptor del cable VCC ajustable externo hacia el lado izquierdo (2,5 V)
- Deslice el interruptor del cable VCC ajustable externo hacia el lado derecho (3,2 V)
- Resultado: el LED naranja responderá parpadeando (0,1 ms, ocho veces) según el código resaltado a continuación porque se detectó el restablecimiento de BOR.

```

else if(MCUSR & 4) { //BOR reset detected
    MCUSR=0;
    for (i=0;i<8;i++)
    {
        //LED OFF
        PORTB &= ~(1 << PINB5); // Set PORTB5 Low
        _delay_ms(100);
        //LED ON
        PORTB |= (1 << PINB5); // Set PORTB5 On
        _delay_ms(100);
    }
}
else if(MCUSR & 8) { // WDT reset detected
    MCUSR=0;
    WDT_Init();
    for (i=0;i<8;i++)
    {
        wdt_reset();
        //LED OFF
        PORTB &= ~(1 << PINB5); // Set PORTB5 Low
        _delay_ms(20);
        //LED ON
        PORTB |= (1 << PINB5); // Set PORTB5 On
        _delay_ms(80);
    }
}

```

(/local--files/8avr:resetexample/step13.png)

10 Tarea 10 - Prueba WDT

- Presione el botón en el tablero durante aproximadamente 1 segundo y luego suelte el botón. Se activarán retrasos prolongados en la rutina principal, lo que provocará un tiempo de espera de WDT.

Nota: El temporizador WDT está configurado en 2 segundos.

```

if((PINB&1<<PINB7)==0){
    PORTB |= (1 << PINB5); // Set PORTB5 high
    _delay_ms(500);
    _delay_ms(500);
    _delay_ms(500);
    _delay_ms(500);
    _delay_ms(500);
    _delay_ms(500);
    _delay_ms(500);
    _delay_ms(500);
}
wdt_reset();
}
}

```

(/local--files/8avr:resetexample/step14.png)

- Resultado: el LED naranja responderá con un parpadeo rápido (0,02 ms encendido, 0,08 ms apagado, cuatro veces) al principio cuando el WDT interrumpe el ISR:

```

//Watchdog timeout ISR
ISR(WDT_vect)
{
    //Burst of 0.1Hz pulses
    for (uint8_t i=0;i<4;i++)
    {
        //LED OFF
        PORTB &= ~(1 << PINB5); // Set PORTB5 Low
        _delay_ms(80);
        //LED ON
        PORTB |= (1 << PINB5); // Set PORTB5 On
        _delay_ms(20);
    }
}

```

(/local--files/8avr:resetexample/step15.png)

- Resultado: el LED naranja volverá a responder con un parpadeo rápido (0,02 ms encendido, 0,08 ms apagado, cuatro veces) a medida que se detecta el restablecimiento del WDT:

```

else if(MCUSR & 8) { // WDT reset detected
    MCUSR=0;
    WDT_Init();
    for (i=0;i<8;i++)
    {
        wdt_reset();
        //LED OFF
        PORTB &= ~(1 << PINB5); // Set PORTB5 Low
        _delay_ms(20);
        //LED ON
        PORTB |= (1 << PINB5); // Set PORTB5 On
        _delay_ms(80);
    }
}
MCUSR=0;
}

```

(/local--files/8avr:resetexample/step16.png)

Q Análisis

El proyecto muestra tres formas en que puede ocurrir un reinicio: POR, BOR y WDT Timeout. Cada uno tiene una aplicación única y todos pueden ejecutarse en la misma aplicación. Los ejemplos de código son solo una referencia de cómo se pueden configurar e implementar estos tipos de restablecimientos.

Conclusiones

El proyecto ayuda a explicar cómo funciona el circuito de reinicio dentro del AVR y cómo implementarlo. La sección de código se puede reutilizar en aplicaciones futuras que pueden requerir una estructura de reinicio similar. De ninguna manera es esta la única forma de diseñar restablecimientos en el dispositivo AVR, este es solo un proyecto de muestra simple que ayuda a explicar la operación y le permite aplicar su conocimiento y comprensión de la estructura de restablecimiento a una aplicación específica.