

Ejemplo de sensor de temperatura interna del convertidor analógico a digital (ADC) AVR de 8 bits

🎯 Objetivo

Este proyecto práctico muestra un ejemplo simple de lectura del sensor de temperatura en el chip. Leer el sensor de temperatura puede ser un proyecto gratificante en sí mismo. Confirma que completó la compilación del software y la configuración del hardware del ADC, y pudo programar el microcontrolador con éxito. Finalmente, el depurador se usa para ver la temperatura usando la ventana de salida de Studio 7.

El sensor de temperatura en el chip está acoplado a un canal ADC8 de un solo extremo. Seleccionar el canal ADC8 escribiendo `ADMUX.MUX[3:0]` a '1000' habilita el sensor de temperatura. La referencia de voltaje interna de 1,1 V también debe seleccionarse para la fuente de referencia de voltaje ADC en la medición del sensor de temperatura. Cuando el sensor de temperatura está habilitado, el convertidor ADC se puede usar en modo de conversión simple para medir el voltaje sobre el sensor de temperatura.

La sensibilidad de voltaje es de aproximadamente 1 mV/°C, la precisión de la medición de temperatura es de ±10°C.

✅ 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	🖥️ Ventanas	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	🖥️ Ventanas
Archivo fuente Main.c	⬇️ (https://microchiptechnology.sharepoint.com/:f:/s/DeveloperHelp/EgvbV6l04yBAvXfl6BemPdoBdqWafKoruX9pkt21RgFG-w?e=pCb004)

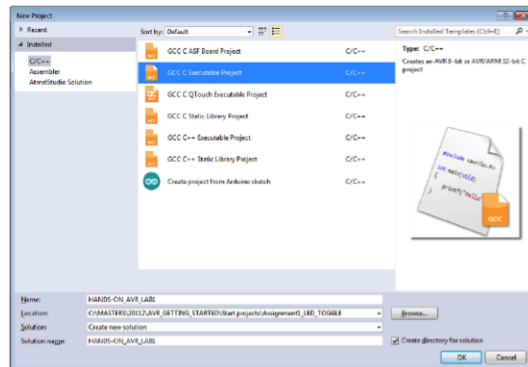
Archivos adicionales

archivos
 Guía del usuario del tablero explicado (http://www.atmel.com/Images/Atmel-42287-ATmega328P-Xplained-Mini-User-Guide_UserGuide.pdf)

Procedimiento

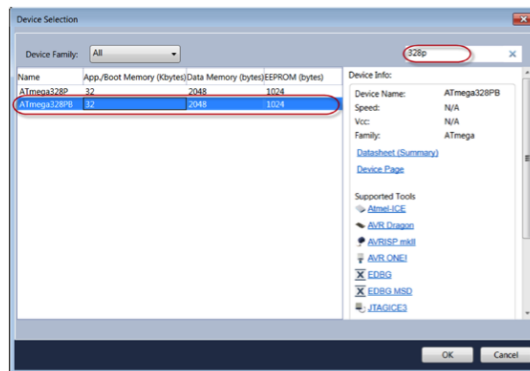
1 Tarea 1: creación de proyectos

- Abrir Atmel Studio 7
- Seleccione **Archivo > Nuevo > Proyecto**
- Seleccione Proyecto ejecutable GCC C y asígnele el nombre **Proyecto1**
- Elija una ubicación para guardar el proyecto en su computadora



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

- Aparecerá la ventana Selección de dispositivo. En la barra de búsqueda, ingrese **328P**, luego seleccione el dispositivo **Atmega328PB** y haga clic en Aceptar.



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

2 Tarea 2 - Main.c

Este proyecto lee el sensor de temperatura interno, convierte el resultado a grados centígrados y luego almacena el resultado en **ADC Temperature Result**.

1) El archivo `main.c` es donde se agrega el código de la aplicación. El proyecto tiene un archivo `main.c` ya creado pero solo contiene una instrucción `while(1)`. Modifique `main.c` ingresando las líneas en el bloque de código gris a continuación.



`#include <avr/io.h>` se agrega automáticamente al archivo `main.c` cuando se genera. Esto siempre debe colocarse antes del bucle principal (vacío). El archivo de encabezado `io.h` llama al archivo `iom328pb.h` que define las definiciones de registro ADC.

```

unsigned int Ctemp;
unsigned int Ftemp;

int main(void)
{
    /* Setup ADC to use int 1.1V reference
    and select temp sensor channel */
    ADMUX = (1<<REFS1) | (1<<REFS0) | (0<<ADLAR) | (1<<MUX3) | (0<<MUX2) | (0<<MUX1) | (0<<MUX0);

    /* Set conversion time to
    112usec = [(1/(8Mhz / 64)) * (14 ADC clocks per conversion)]
    and enable the ADC*/
    ADCSRA = (1<<ADPS2) | (1<<ADPS1) | (1<<ADEN);

    /* Perform Dummy Conversion to complete ADC init */
    ADCSRA |= (1<<ADSC);

    /* wait for conversion to complete */
    while ((ADCSRA & (1<<ADSC)) != 0);

    /* Scan for changes on A/D input pin in an infinite loop */
    while(1)
    {
        /* start a new conversion on channel 8 */
        ADCSRA |= (1<<ADSC);

        /* wait for conversion to complete */
        while ((ADCSRA & (1<<ADSC)) != 0)
        ;

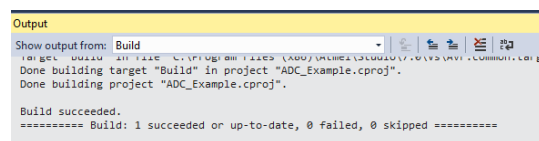
        /* Calculate the temperature in C */
        Ctemp = (ADC - 247)/1.22;
        Ftemp = (Ctemp * 1.8) + 32;
    }

    return -1;
}

```

3 Tarea 3 - Proyecto de construcción

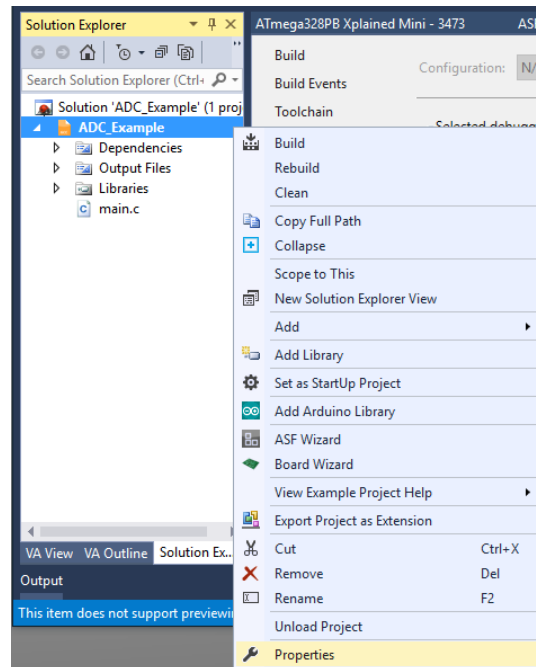
- Seleccione **Build > Build Solution** en el menú de Studio 7 para compilar el código. Verá un mensaje de **Build Succeeded** en la ventana de resultados. Si hay algún error, verifique `main.c` para ver si hay errores al ingresar el código del programa.



(/local--files/8avr:avradc/output.png)

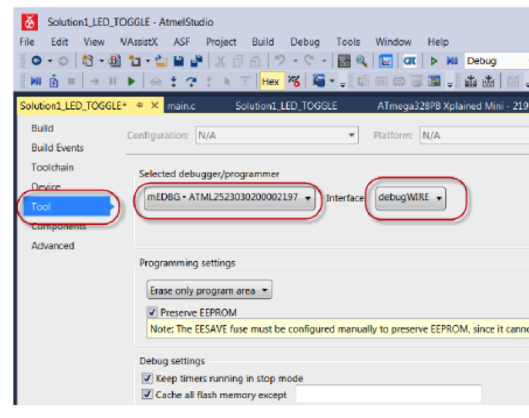
4 Tarea 4: programación de la placa explicada

- Conecte la placa Xplained al puerto USB de la computadora usando el cable incluido.
- En el área **Explorador de soluciones**, haga clic con el botón derecho en el nombre del proyecto y seleccione **Propiedades**.



(/local--files/8avr:avradc/solution.png)

- En la selección del menú **Herramienta**, elija **mEDBG** y **debugWire** como interfaz.



(/local--files/8avr:avradc/step42.png)

- Selecione **Depurar > Iniciar sin depurar** en el menú de Studio 7. El proyecto construirá y luego programará el tablero explicado con el código del proyecto junto con el control de depuración.



(/local--files/8avr:avradc/step52.png)

- Studio 7 mostrará un mensaje **Listo cuando se complete la programación**.

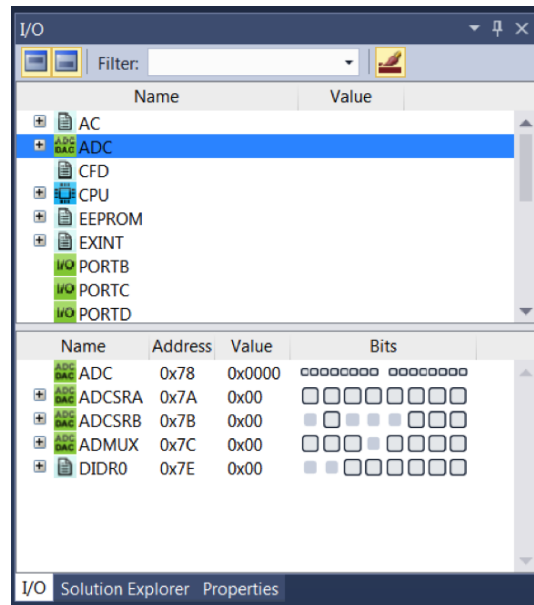


Si la placa explicada no se conecta, es posible que haya un ajuste de fusible que provoque que esto ocurra (/boards:debugbrick).

5 Tarea 5 - Depuración

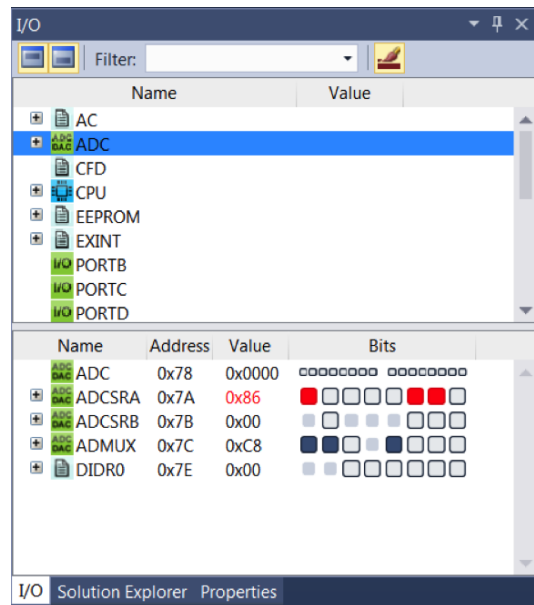
La depuración de un dispositivo es esencial para determinar cómo se puede ejecutar un programa.

- Selecione **Depurar > Iniciar depuración y romper**. El proyecto se compilará y el programa se cargará en el tablero Xplained
- Se abrirá la ventana I/O View mostrando los diversos periféricos
- Haga clic en la selección de **ADC** para abrir la vista de E/S para el ADC



(/local--files/8avr:avradc/adcdebug.png)

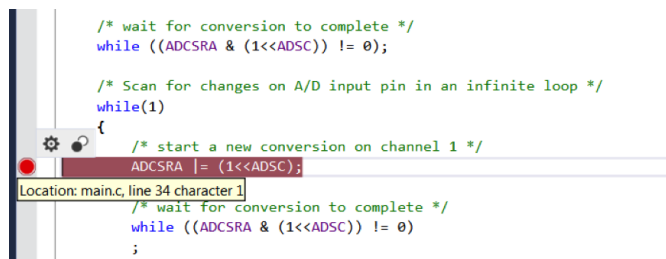
- Seleccione **Depurar > Pasar por alto (F10)** para recorrer el programa en un solo paso en el tablero Xplained. Supervise los registros ADC en la vista de E/S mientras ejecuta un solo paso



(/local--files/8avr:avradc/adcdebug2.png)

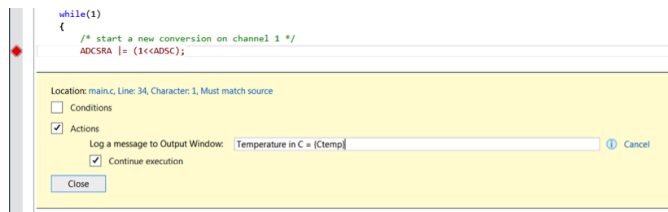
6 Tarea 6: punto de interrupción y salida

- Haga clic en **Depurar > Romper todo** en el menú superior de Studio 7
- Haga clic en el margen para habilitar un punto de interrupción en la línea de comando en la instrucción ADCSRA dentro del ciclo While



(/local--files/8avr:avradc/breakpoint.png)

- Mueva el mouse sobre el círculo rojo del punto de interrupción para poder ver la opción emergente de configuración (símbolo de engranaje) y haga clic en él

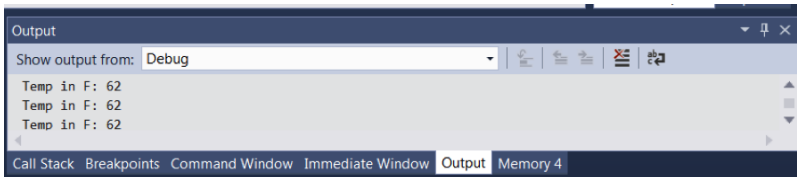


(/local--files/8avr:avradc/breakpoint2.png)

- Dentro de la **Ventana** de configuración de punto de interrupción , marque la casilla **Acciones** . Dentro de "Registrar un mensaje en la ventana de salida", inserte lo siguiente: **Temperatura en C = {Ctemp}** y marque la **casilla Continuar ejecución** . Haga clic en **Cerrar**
- Ingrese al modo de depuración haciendo clic en **Iniciar depuración y romper** en el menú de **depuración superior**
- Abra la **ventana de salida** seleccionando **Depurar > Ventanas > Salida** para ver el valor de la variable de temperatura
- Haga clic en **Depurar > Continuar** y vea la temperatura en la ventana Salida

✱ Resultados

La temperatura del chip se muestra en la ventana de salida. Al presionarlo con un dedo, la lectura se calentará un par de grados.



(/local--files/8avr:avradc/breakpoint3.png)

7 Tarea 7: deshabilitar debugWire y cerrar

El **fusible debugWire** debe restablecerse para poder programar la placa Xplained en el futuro. Mientras aún se ejecuta en modo de depuración, seleccione **Depurar > Deshabilitar debugWire y Cerrar** . Esto liberará el fusible debugWire.

Q Análisis

Usar el ADC es bastante fácil y la función de depuración facilita el seguimiento de los resultados.

💡 Conclusiones

Este proyecto puede convertirse en la base para futuros proyectos relacionados con ADC.