

Programación de Sistemas Embebidos 2020
Facultad de Informática
Universidad Nacional del Comahue
Trabajo Práctico 2

1- Escribir un programa hello world en C, que haga parpadear el led conectado en la placa pro mini. Puede utilizar el código del apunte “El primer programa embebido” aunque este puede contener errores (está advertido).
Utilice el esquemático de la placa para reconocer cuál es el pin en el microcontrolador AVR al que está conectado el led.

Recursos

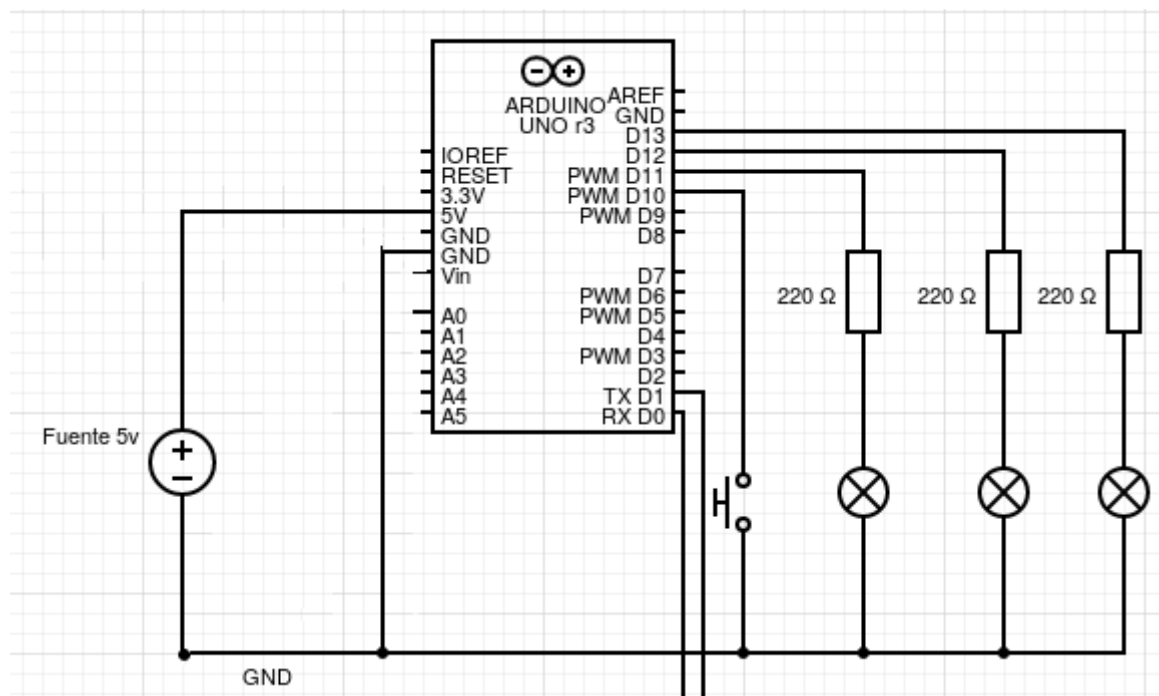
- Utilice el makefile que se provee con este TP en el source code de ejemplo:
<https://github.com/zrafa/pse2020/tree/master/TPs/tp2-pse>
- Clase del primer programa embebido
<http://se.fi.uncoma.edu.ar/pse2020/referencia/Clase-hello-world.pdf>
- Demás recursos de la materia:
<http://se.fi.uncoma.edu.ar/pse2020/>

Responda:

1. ¿A que pin de la placa pro mini está conectado el led?
2. ¿A qué pin del microcontrolador AVR está conectado el led?

2- Crear una segunda aplicación para AVR que controle 4 pines de E/S digital paralela del atmega328p (GPIO).

Hardware: utilice el hardware que se detalla a continuación.



Esquemático creado con : <https://www.circuit-diagram.org/>

El hardware a utilizar es un boton o pulsador para ENTRADA, y 3 LEDs para SALIDA. Todo el hardware debe ser controlado por pines conectados al periférico de E/S GPIO PORTB.

La fotoresistencia y las señales RX Y TX no son necesarias en este TP.

Programación de Sistemas Embebidos 2020
Facultad de Informática
Universidad Nacional del Comahue
Trabajo Práctico 2

Software: El funcionamiento es el siguiente:

- El programa tiene dos modos: el modo cero y el modo uno .
- Cuando inicia está en modo cero, si se presiona el pulsador se cambia de modo (una pulsación cambia a modo uno, una nueva pulsación cambia a modo cero, etc).
- **En modo cero** los 3 LEDs parpadean al mismo tiempo, con una espera de 1 segundo entre parpadeo y parpadeo.
- **En modo uno** el sistema es un contador, de 0 a 7, en base dos. Utilizar los 3 LEDs para ir mostrando los valores binarios. Un LED encendido es un dígito uno en base dos, y un LED apagado es un dígito cero (también en base dos). Entre valor y valor hay una espera de un segundo.

Realice una verificación del programa utilizando avrdude y transfiriendo el programa a la flash del atmega328. Ejecutar. Para realizar este ejercicio es muy útil el capítulo 3 del libro, el esquemático del arduino pro mini y la hoja de datos del atmega328p.

SUGERENCIAS:

Conviene “dividir y conquistar”:

- Escriba una función que devuelva como resultado si el botón está presionado o no está presionado. Puede utilizar una variable global para mantener el modo. Trate de quitar el rebote (debouncing)
- Escriba una función que parpadee los 3 leds.
- Para cada modo escriba una función diferente, de manera de pensar cada modo por separado. Por ej. para el modo contador, es muy útil escribir al puerto que controla los leds el valor binario directamente, en vez de prender los leds “de a uno”. El dispositivo de salida no es mas que un registro binario, donde cada bit controla un led.

3- Utilice avr-objdump para realizar un decodificado del programa binario elf (obtener el código en lenguaje ensamblador AVR a partir del binario elf). Observe ahora que al compilar el programa para un microcontrolador particular aparece al principio del código la tabla de vectores.

Presente en este ejercicio las primeras 10 posiciones de la tabla de vectores.

4- **Modifique el hardware:** quitar el pulsador y colocar los leds en los pines de la placa 10, 11, y 12.

Escriba un programa que realice el efecto del knight-rider, utilizando esos 3 leds y el conectado en la placa en el pin 13 (es decir, utilizando 4 leds). Si cuenta con 5 leds quedará aún mejor.

5- Responder:

- a) ¿Qué hw existe en cada dispositivo de E/S que permite su programación?
- b) ¿Cómo se llaman en el PORT B del AVR?
- c) ¿Cuántos dispositivos de E/S observa conectados al DATABUS del AVR? Mencione sus nombres.
- d) Usted debe controlar un nuevo dispositivo de E/S en un sistema, utilizando un programa escrito en lenguaje C. ¿Que detalles necesita conocer para escribir el software? ¿De dónde obtiene los detalles?
- e) ¿Qué es el voltage? ¿Qué es la corriente?
- f) ¿Por qué se lo clasifica como “arquitectura harvard” a los micros AVR de 8-bits?

Verificación: compilar, vincular y enviar el firmware a cada AVR.

Entrega: subir (push) el trabajo práctico resuelto (o sus versiones intermedias) al repositorio git compartido (<http://github.com/zrafa/pse2020.git>)