

Modificación bootloader

Para poder modificar el tiempo de espera del bootloader para arrancar el programa principal se partió del bootloader de *Sanguino* (makefile + código fuente en c). El hexadecimal obtenido al ejecutar el makefile para ATMEGA644P no es válido porque se sale fuera de la memoria de programa del microcontrolador.

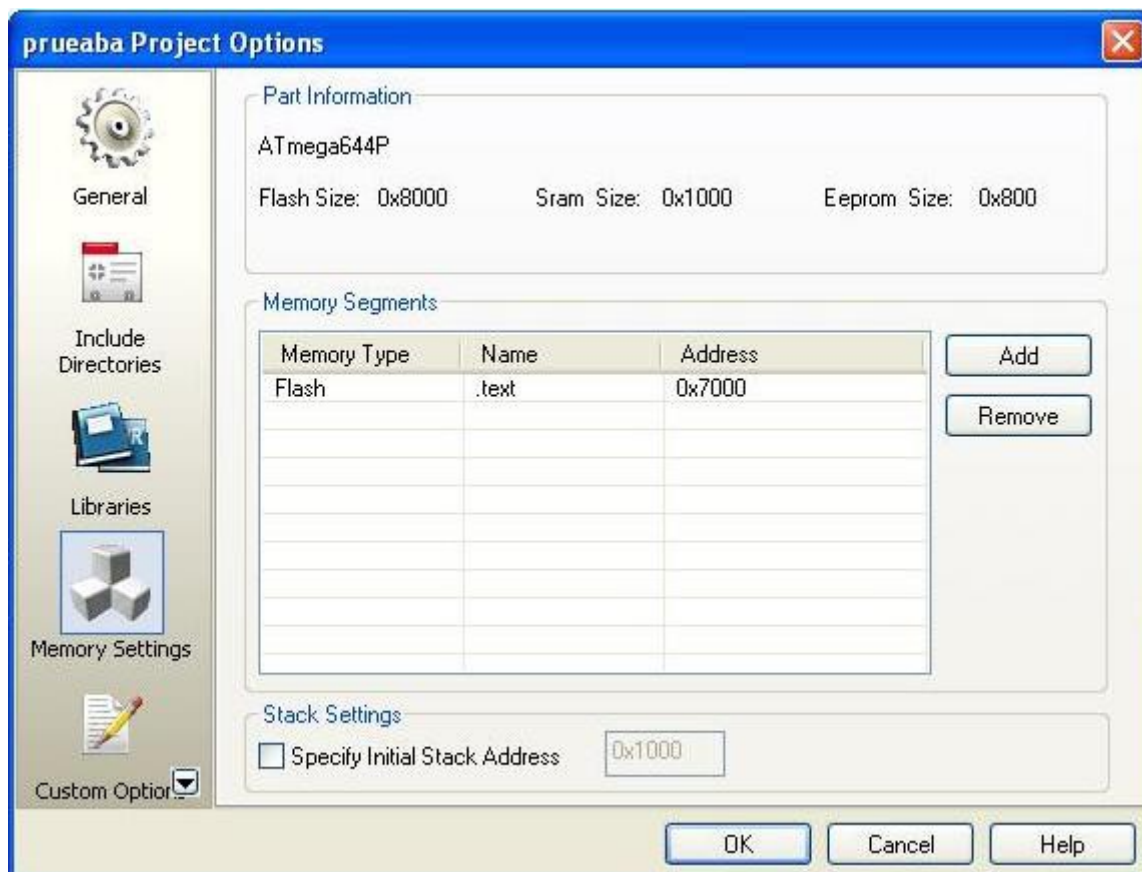
Para obtener un makefile correcto se utilizó el entorno AVR Studio 4. Se creó un proyecto nuevo con el mismo código fuente ofrecido por arduino. De esta manera fue posible compilar el código fuente de forma que entrara en la memoria del micro (desde la dirección de memoria 0).

Hay que definir en el .c algunos defines que se definen en el makefile original y modificar los pines de los leds

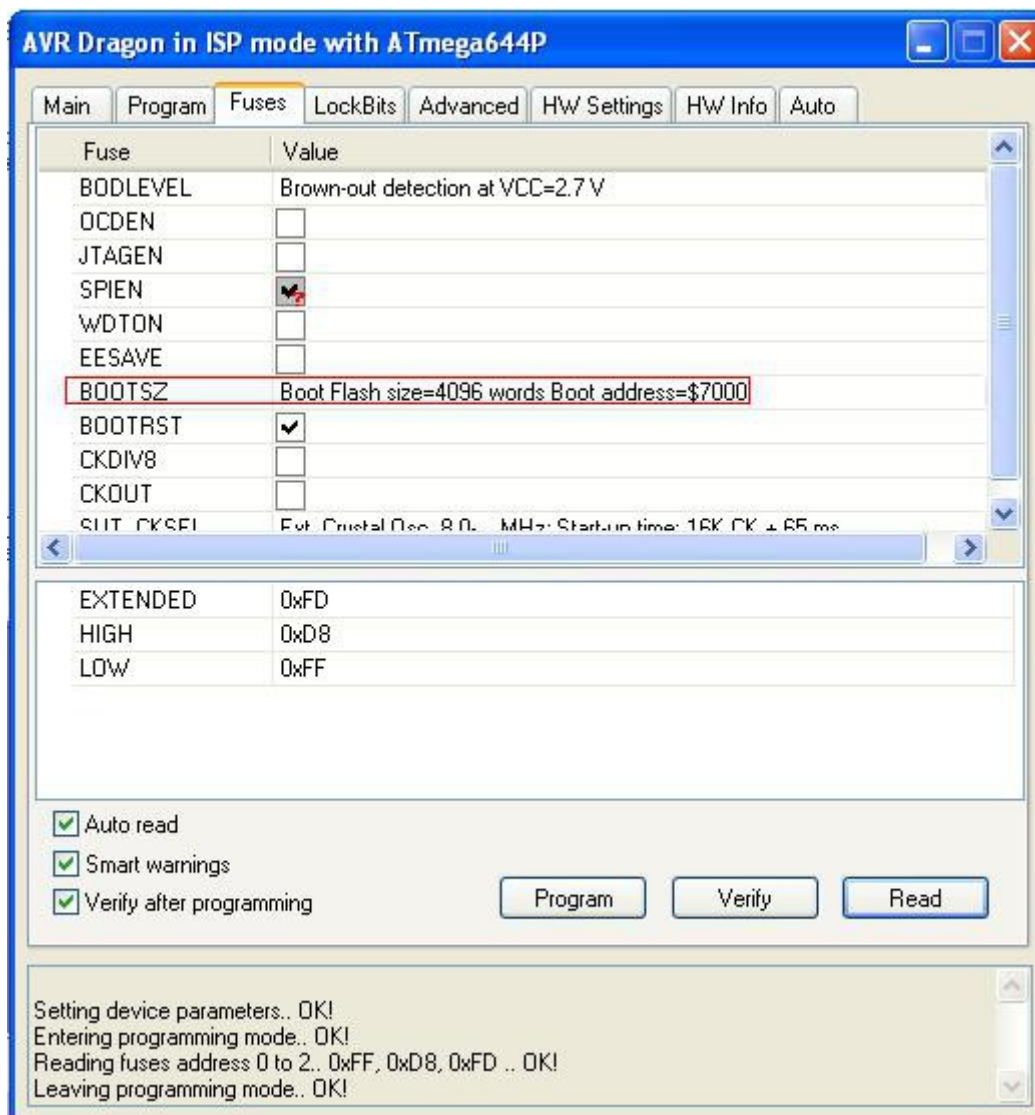
```
#define F_CPU 16000000
#define ADABOOT
#define MAX_TIME_COUNT 8000000L>>1
#define LED_DDR DDRC
#define LED_PORT PORTC
#define LED_PIN PINC
#define LED PINC3
#define LEDGREEN PINC2

#define __AVR_ATmega644P__
```

El microcontrolador tiene definida una dirección de inicio de bootloader. Para que al ejecutar el makefile se cree un hexadecimal con el código en la dirección de inicio de bootloader en el AVR Studio Project/Configuration Options/... en la pestaña memory settings añadir un nuevo registro.



Hay que modificar también las FUSES para definir la dirección de inicio del bootloader al micro.



Con esto se compila el programa, se puede programar en el micro y cargar cualquier programa mediante Arduino. Pero no se corrige el problema del retardo. Para ello se quitan la secuencia de encendido de leds y se modifica la temporización del timer a 0,5 s. Es necesario resetear el watchdog en algunos comandos para que no se produzca un reset cuando se está cargando el programa con arduino.