

Incorporación de bootloaders a los PWRC2 / INTI-NQN 2021 para Laboratorio Cutral-Co L&R Ingeniería – Rev. 2 08-03-2021 R. Oliva

1. RESUMEN

1.1 Incorporación de Bootloader a placas CL2 de los equipos PWRC2. Se continuó el trabajo iniciado en 2020, utilizando herramientas abiertas, para incorporar a las placas un mecanismo de bootloader por puerto serie a 115200 baud. Para ello se realizó la adaptación de dicha placa (que es contemporánea pero no nativamente "Arduino compatible") para su utilización con el Arduino IDE [ref0] y la función "quemar bootloader", utilizando una versión modificada de Optiboot de MightyCore [ref1] y el programa Avrdude [ref2]. Si bien la documentación y procedimiento debe depurarse, esto permite como se muestra en la Figura 1 acceder a las placas CL2, grabar el programa Optiboot en la zona especialmente reservada (accedida previamente al arranque del programa principal), con el clásico programador AVRISP Mkii con que cuenta el Laboratorio de INTI-Cutral Có.

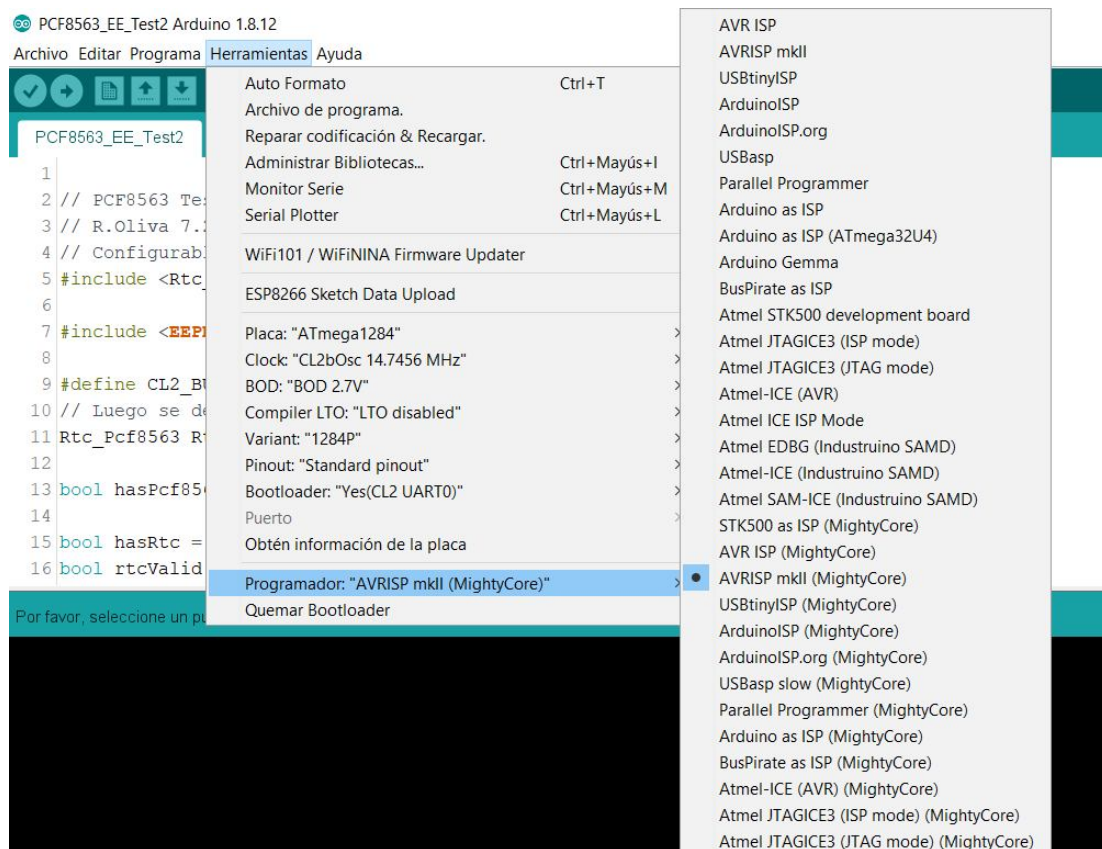


Figura 1 – Acceso previo al PWRC2 para grabación del bootloader Optiboot [ref1], utilizando ArduinoIDE

Como se observa, se utiliza una adaptación de las placas genéricas MightyCore para que CL2b aparezca en el menú, con la configuración de reloj, fusibles y demás parámetros específicos de dicha placa. Esta adaptación se irá mejorando para que resulte más directo el proceso de grabación del bootloader en todas las unidades PWRC2. Es importante hacer backup de Flash y EEPROM ya que la operación borra la memoria completa del controlador ATmega1284P [ref3]. Asimismo, se debe seleccionar AVRISP mkii (MightyCore), y recordar extraer la SD del equipo (como en los updates de Firmware) porque interfiere con la programación via el puerto SPI.

1.2 Utilización del Bootloader en placas CL2 / PWRC2. Una vez finalizada la incorporación Optiboot, se apaga el PWRC2 y se retira el AVRISP, ya que la programación de Firmware ahora puede hacerse directamente vía puerto serie y a mayor velocidad (en forma similar a una placa Arduino Uno o Mega). Ahora solamente queda en memoria Flash el bootloader, no aparecerá display si encendemos nuevamente el PWRC2 el equipo queda esperando la carga de nuevo firmware con el protocolo STK500, y el OLED parpadea con 5 OLED blinks / seg, como se muestra en Figura 2. Aquí el ensayo es sobre un equipo similar (SISMED/SJ24) a los PWRC2 del INTI, y se supone que está conectado un convertor Serie/USB al COM principal del PWRC2 y conectado a la PC.



Figura 2 reencendido con bootloader Optiboot cargado, a la espera de nuevo Firmware vía puerto serie

Se copia el firmware original (aquí main1284.hex) realizado en C por el compilador Codevision AVR [ref1b], para el equipo SISMED/SJ24 (Figura 3), y se copia además el archivo send2.bat, que cumple funciones similares a lo que realiza internamente el Arduino IDE (llamando al programa Avrdude con una determinada configuración) para programar las placas basadas en procesador AVR. Además, se invoca la línea de comandos (cmd.exe) tanto en Win7 como en Win10, en el directorio en que se está trabajando, y se escribe “send2 main1284.hex”, todavía sin presionar Enter:

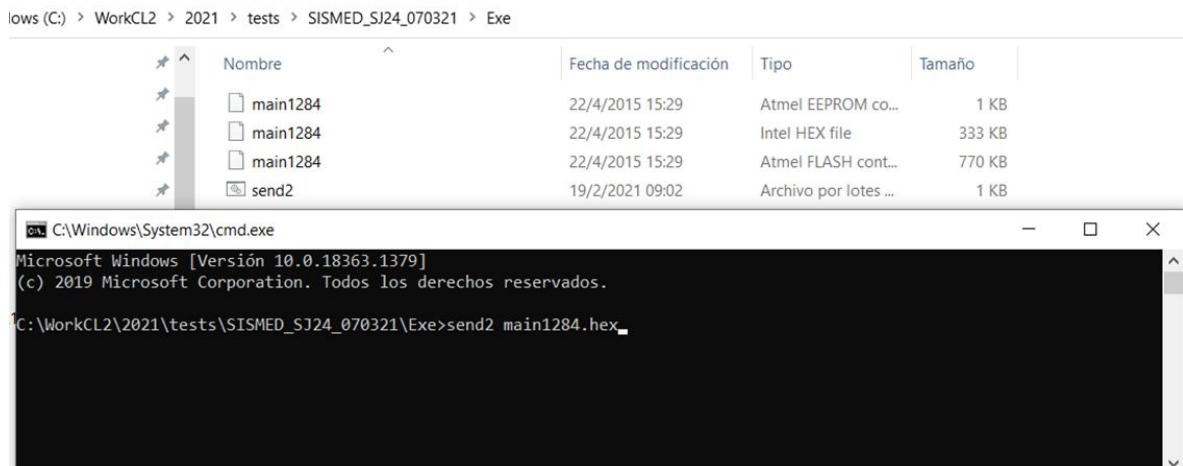


Figura 3 Firmware main1284.hex y archivo send2.bat copiados en el mismo directorio, y llamado a send2.



A continuación se presiona Enter, y después (pero casi en simultáneo) se enciende el PWRC2, con lo cual se debería comunicar el programa Avrdude que llama send2.bat con el Optiboot cargado en la placa CL2, e iniciar una secuencia de lectura/escritura/verificación como se muestra en la Figura 4.

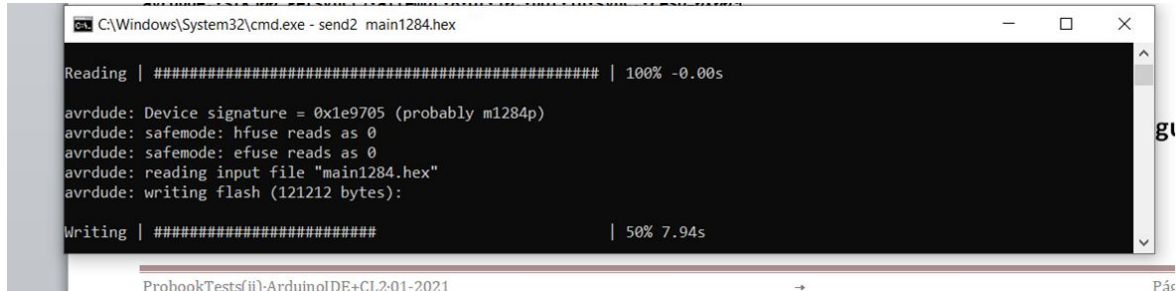


Figura 4 Ejecución correcta de send2.bat para grabar el firmware nuevo en el PWRC2

A continuación se muestra un extracto de texto de la ventana de comandos, y la secuencia que ejecuta el avrdude para realizar la programación del equipo PWRC2:

C:\WorkCL2\2021\tests\SI SMED_SJ24_070321\Exe>send2 mai n1284. hex

C:\WorkCL2\2021\tests\SI SMED_SJ24_070321\Exe>C:\Users\Rafael\AppData\Local\Ardui no15\packages\ardui no\ tool s\avrdude\6.3.0-ardui no8\bi n\avrdude -CC:\Users\Rafael\AppData\Local\Ardui no15\packages\Mi ghtyCore\hardware\avr\2.0.3\avrdude.conf -v -patmega1284p -cardui no -PCOM8 -b115200 -D -Ufl ash:w: mai n1284. hex: i

avrdude: Version 6.3, compiled on Dec 16 2016 at 13:33:19
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
Copyright (c) 2007-2014 Joerg Wunsch

System wide configuration file
"C:\Users\Rafael\AppData\Local\Ardui no15\packages\Mi ghtyCore\hardware\avr\2.0.3\avrdude.conf"

```
Using Port           : COM8
Using Programmer     : ardui no
Overriding Baud Rate : 115200
AVR Part             : ATmega1284P
Chip Erase delay     : 55000 us
PAGEL                : PD7
BS2                  : PA0
RESET di spositi on : dedicated
RETRY pulse         : SCK
serial program mode  : yes
parallel program mode : yes
Timeout              : 200
StabDelay            : 100
CmdexeDelay          : 25
SyncLoops            : 32
ByteDelay            : 0
Poll Index           : 3
Poll Value           : 0x53
Memory Detail       :
```

Memory	Type	Mode	Block Delay	Block Size	Poll Indx	Paged	Block Size	Page Size	#Pages	MinW	MaxW	Poll ed ReadBack
ee prom		65	10	128	0 no		4096	8	0	9000	9000	0xff 0xff
fl ash		65	10	256	0 yes		131072	256	512	4500	4500	0xff 0xff
l ock		0	0	0	0 no		1	0	0	9000	9000	0x00 0x00
l fuse		0	0	0	0 no		1	0	0	9000	9000	0x00 0x00
h fuse		0	0	0	0 no		1	0	0	9000	9000	0x00 0x00
efuse		0	0	0	0 no		1	0	0	9000	9000	0x00 0x00
si gnature		0	0	0	0 no		3	0	0	0	0	0x00 0x00
cali bration		0	0	0	0 no		1	0	0	0	0	0x00 0x00



Programmer Type : Arduino
Description : Arduino
Hardware Version: 3
Firmware Version: 8.0
Vtarget : 0.3 V
Varef : 0.3 V
Oscillator : 28.800 kHz
SCK period : 3.3 us

avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% -0.00s

avrdude: Device signature = 0x1e9705 (probably m1284p)
avrdude: safemode: hfuse reads as 0
avrdude: safemode: efuse reads as 0
avrdude: reading input file "main1284.hex"
avrdude: writing flash (121212 bytes):

Writing | ##### | 100% 15.91s

avrdude: 121212 bytes of flash written
avrdude: verifying flash memory against main1284.hex:
avrdude: load data flash data from input file main1284.hex:
avrdude: input file main1284.hex contains 121212 bytes
avrdude: reading on-chip flash data:

Reading | ##### | 100% 11.61s

avrdude: verifying ...
avrdude: 121212 bytes of flash verified

avrdude: safemode: hfuse reads as 0
avrdude: safemode: efuse reads as 0
avrdude: safemode: Fuses OK (E:00, H:00, L:00)

avrdude done. Thank you.

Posteriormente, se apaga el PWRC2 y se vuelve a insertar la SD, y se abre el clásico Hyperterminal o Teraterm a 19200 baud y la secuencia es la habitual al encender el PWRC2 (ver listado y Figura 5):

```
Probando v1284P-SISM2_SJ24 15-03-2015
LCD Access testing.
-TestCtr=1 -TestCtr=2 -TestCtr=3 -TestCtr=4 -TestCtr=5 -TestCtr=6
Ahora RTC test..
RTC inicializando..
07/03/2021-21:16:05
Set CLKOUT to 1Hz
```

Ahora PSoC-M4E3 prueba de acceso (Read)..

```
M4EReadStatus_in = 0
M4EReadStatus_after = 0
bVect = 0
bPW_OK_En = 1
bBuzz = 0
bAux2 = 0
bLED = 0
bK_Aux = 0
IN0-P27 = 1
IN1-P25 = 0
IN2-P23 = 0
IN2: IN0 = 1
IN3-P21 = 0
IN4-P13 = 1
IN4: IN3 = 2
```

IN14-P20 = 1
IN15-P22 = 1
IN16-P24 = 0

.... (partes eliminadas, requiere reconfiguración)..

PreLg Archivo de Eventos: EA070321.CSV
PostLg Archivo de Eventos: EA070321.CSV
PW2 GIStatus find + Set Measurements to Start..
PW2 a) FRP Status: 3
b) IAE_DIPSw: 1
c) VBat_DIPSw: 2
d) Global Status: 3

Archivo de Eventos: EA070321.CSV
SISM2 - Inicializaci3n Curva Potencia (Presione ? para comandos):
? - Listar comandos.
1 - Menu de Setup Prueba
2 - Menu Setup Extendido
3 - Lanzar Prueba c/param. actuales (nuevo archivo)
4 - Re-Lanzar Prueba (archivo en curso)
5 - Configurar Canales

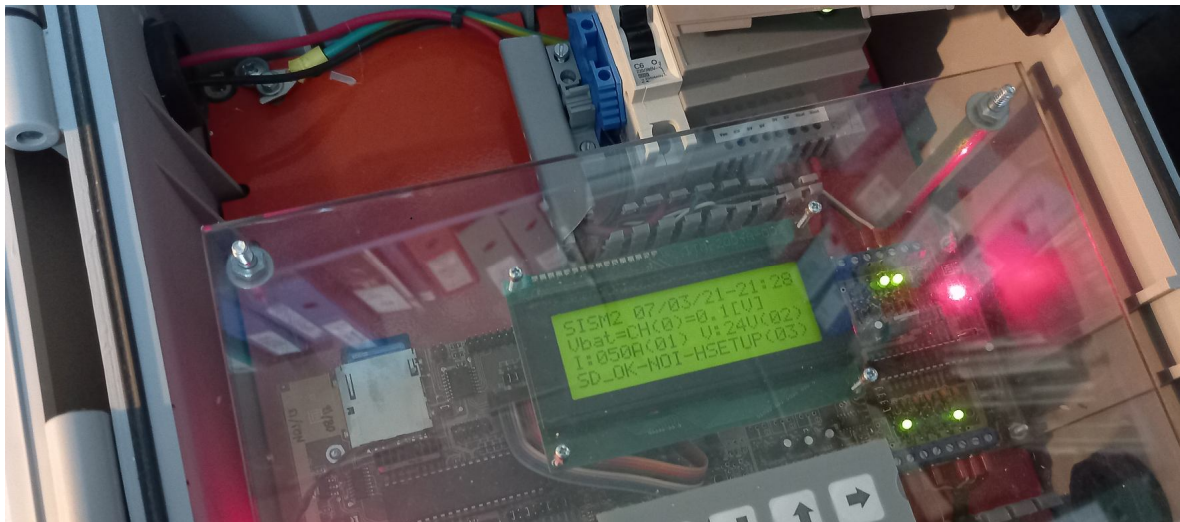


Figura 5 Vista de ejecuci3n una vez programado del main1284.hex en un equipo SISMED similar al PWRC2, 7/3/21

1.3 Ventajas del Bootloader en placas CL2 / PWRC2. Se observa que el procedimiento facilita la actualizaci3n del firmware, que hasta el momento requería la remoci3n de la tapa transparente del PWRC2, la conexi3n del programador AVRISP/USBASP y la utilizaci3n de AVRStudio4 o similar para realizar los upgrades de Firmware, que habitualmente tardaba varios minutos. A trav3s del Optiboot, y una vez que se concluya el proceso de mejora de parámetros de configuraci3n e interfaz gráfica, el proceso como se ve del extracto se concluye en menos de 30 segundos.

Por otro lado, la utilizaci3n de la combinaci3n CL2 y Optiboot se ha ensayado para la plataforma Arduino IDE más tradicional (que utiliza el lenguaje de programaci3n simplificado (como se ve en el fondo de Figura 1). Sobre esto se está construyendo una nueva versi3n de PWRC2 abierta / OpenDLogger, de menor complejidad (el firmware actual del PWRC2 es de un tamaño medio pero supera las 41000 líneas de código C), que utilice programas clásicos (ejemplo test.ino), con sencillas secuencias setup() y loop(), además de muchas de las bibliotecas accesibles libremente de Arduino e implemente algunas propias que faciliten el acceso a los periféricos estándar de los PWRC2. Esto facilitará los ensayos muy específicos pero de corta duraci3n y que puedan ser programados directamente por los ingenieros del INT1 y modificados con cierta agilidad en comparaci3n con la versi3n anterior.

La reciente adquisici3n en 2020 de Atmel (desarrollador de la línea de microcontroladores AVR) por parte de Microchip y la clara línea de sostenimiento de los modelos utilizados ATmega [ref3], que viene mostrando la empresa hace que resulte relevante

sostener la actualización de este tipo de equipos, sobre todo en las actuales circunstancias económicas locales, por su costo reducido y disponibilidad de repuestos.

Asimismo se ha ensayado el uso de Optiboot/CL2 con la herramienta abierta PlatformIO [ref4], que facilita la programación en C++ en forma directa para muchos microcontroladores (además de la línea AVR, por ejemplo ESP8266 o ESP32) y se configura como plug-in en editores más avanzados como Atom o Visual Studio Code (VSC [ref5], gratuito) . En la Figura 6 se aprecia el ensayo bajo Linux / Ubuntu 18.04 de una placa CL2 con la biblioteca LiquidCrystal de Arduino, utilizando VSC con el plug-in PlatformIO, y un display alfanumérico reducido de 16 x 2.

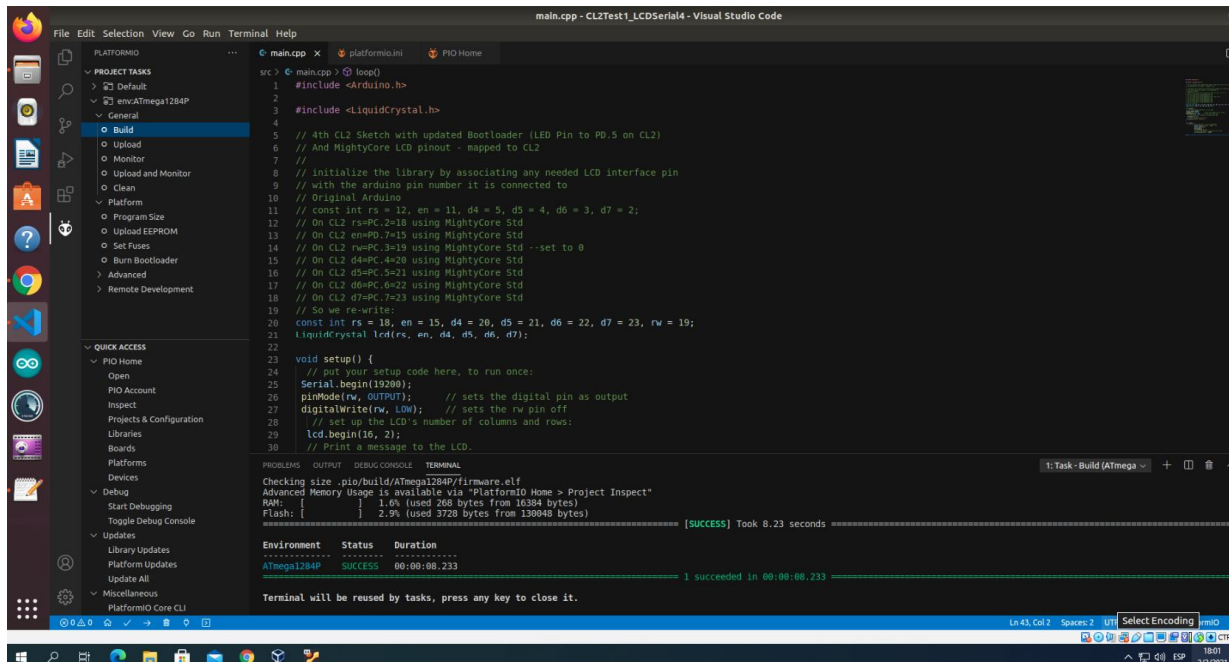


Figura 6 Utilización de Visual Studio Code y el plug-in PlatformIO bajo Linux con una placa CL2 y Optiboot conectada (02-21)

-0-

1.4 Referencias

- [ref0] <https://www.arduino.cc/en/software>
 [ref1] <https://github.com/MCUdude/MightyCore>
 [ref1b] <https://www.codevision.be/>
 [ref2] <https://www.nongnu.org/avrdude/>
 [ref3] <https://www.microchip.com/wwwproducts/en/ATmega1284>
 [ref4] <https://platformio.org/>
 [ref5] <https://code.visualstudio.com/>

Fecha de Revisión: Marzo 8 - 2021

NOTAS: