.cpu arm7tdmi

.eabi\_attribute 20, 1

.eabi\_attribute 21, 1

.eabi\_attribute 23, 3

.eabi\_attribute 24, 1

.eabi\_attribute 25, 1

.eabi\_attribute 26, 1

.eabi\_attribute 30, 6

.eabi\_attribute 34, 0

.eabi\_attribute 18, 4

.file "scu\_set.c"

.text

;declaracion de funcion “SCU\_SetPin”

;void SCU\_SetPin(SCU\_T \*pSCU, unsigned char port, unsigned char pin, unsigned char function)

;{ pSCU->SFSP)[port][pin]|= function | (2<<3); }

.align 2

.global SCU\_SetPin

.syntax unified

.arm

.fpu softvfp

.type SCU\_SetPin, %function

SCU\_SetPin:

@ Function supports interworking.

@ args = 0, pretend = 0, frame = 8

@ frame\_needed = 1, uses\_anonymous\_args = 0

@ link register save eliminated.

str fp, [sp, #-4]!

add fp, sp, #0

sub sp, sp, #12

str r0, [fp, #-8] ;guarda el puntero pSCU

mov r0, r1

mov r1, r2

mov r2, r3

mov r3, r0

strb r3, [fp, #-9] ;guarda el argumento unsigned char port

mov r3, r1

strb r3, [fp, #-10] ;guarda el argumento unsigned char pin

mov r3, r2

strb r3, [fp, #-11] ;guarda el argumento unsigned char function

ldrb r0, [fp, #-9] @ zero\_extendqisi2 ;carga variable port en r0

ldrb r2, [fp, #-10] @ zero\_extendqisi2 ;carga variable pin en r2

ldrb ip, [fp, #-9] @ zero\_extendqisi2 ;carga variable port en ip

ldrb r1, [fp, #-10] @ zero\_extendqisi2 ;carga variable pin en r1

ldr r3, [fp, #-8] ;carga el puntero pSCU en r3

lsl ip, ip, #5 ;ip=ip<<5 ⇒ ip=port<<5

add r1, ip, r1 ;r1=r1+ip ⇒ r1=pin+port<<5

ldr r3, [r3, r1, lsl #2] ;carga en r3 la direccion de pSCU->SFSP[port][pin]

;SFSP corresponde al primer campo

ldrb r1, [fp, #-11] ;carga function en r1

;function | (2<<3)

orr r1, r1, #16 ;#16 = (2<<3)

and r1, r1, #255

;or y asignación → |=

orr r1, r3, r1

ldr r3, [fp, #-8]

lsl r0, r0, #5

add r2, r0, r2

str r1, [r3, r2, lsl #2] ;carga y guarda el nuevo valor de (pSCU->SFSP)[port][pin] nop

add sp, fp, #0

@ sp needed

ldr fp, [sp], #4

bx lr

.size SCU\_SetPin, .-SCU\_SetPin

;declaracion de funcion “SCU\_SetEZI”

;void SCU\_SetEZI(SCU\_T \*pSCU, unsgn char port, unsgn char pin, unsgn char ezi) ;{(pSCU->SFSP)[port][pin]|=(1<<ezi);}

;funciona similar a la función SCU\_SetPin

.align 2

.global SCU\_SetEZI

.syntax unified

.arm

.fpu softvfp

.type SCU\_SetEZI, %function

SCU\_SetEZI:

@ Function supports interworking.

@ args = 0, pretend = 0, frame = 8

@ frame\_needed = 1, uses\_anonymous\_args = 0

push {fp, lr}

add fp, sp, #4

sub sp, sp, #8

str r0, [fp, #-8]

mov r0, r1

mov r1, r2

mov r2, r3

mov r3, r0

strb r3, [fp, #-9]

mov r3, r1

strb r3, [fp, #-10]

mov r3, r2

strb r3, [fp, #-11]

ldr ip, .L3

ldrb r1, [fp, #-9] @ zero\_extendqisi2

ldrb r3, [fp, #-10] @ zero\_extendqisi2

ldr lr, .L3

ldrb r0, [fp, #-9] @ zero\_extendqisi2

ldrb r2, [fp, #-10] @ zero\_extendqisi2

lsl r0, r0, #5

add r2, r0, r2

ldr r2, [lr, r2, lsl #2]

ldrb r0, [fp, #-11] @ zero\_extendqisi2

mov lr, #1

lsl r0, lr, r0

orr r2, r2, r0

lsl r1, r1, #5

add r3, r1, r3

str r2, [ip, r3, lsl #2]

nop

sub sp, fp, #4

@ sp needed

pop {fp, lr}

bx lr

.L4:

.align 2

.L3:

.word 1074290688

.size SCU\_SetEZI, .-SCU\_SetEZI

.ident "GCC: (15:6.3.1+svn253039-1build1) 6.3.1 20170620"