.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 "leds\_setup.c"

;int toggle=0;

.global toggle

.bss

.align 2

.type toggle, %object

.size toggle, 4 ;tamaño 4 por ser un int

toggle:

.space 4

.text

;declaracion funcion LED\_cfg

;void LED\_cfg(led\_t led);

.align 2

.global LED\_cfg

.syntax unified

.arm

.fpu softvfp

.type LED\_cfg, %function

LED\_cfg:

@ Function supports interworking.

@ args = 0, pretend = 0, frame = 16

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

push {fp, lr} ;guarda el “frame pointer” y el “Link register” en el stack

add fp, sp, #4

;port,pin,port\_gpio,pin\_gpio,function;

sub sp, sp, #16 ;crea el espacio para las variables locales

mov r3, r0 ;copia lo que viene como argumento de la funcion (led) en r3

strb r3, [fp, #-13] ;se guarda r3 en la posicion fp-13, es decir fp-13 representa a

;la variable led

;port=2

mov r3, #2 ;un 2 en r3

strb r3, [fp, #-9] ;se guarda en la posicion fp-9 r3

;es decir que ahora fp-9 representa port

;if(led==L0R || led==L0G || led==L0B)

;led==L0R → comparacion contra 0

ldrb r3, [fp, #-13] @ zero\_extendqisi2 ;carga fp-13 que corresponde a la variable led

cmp r3, #0

beq .L2 ;salto a .L2 si resultado de cmp es true → entra al if

;led==L0G → comparacion contra 1

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #1

beq .L2 ;salto a .L2 si resultado de cmp es true → entra al if

;led==L0B

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #2

bne .L3 ;salto a .L3 si resultado de cmp es false →siguiente else if

.L2: ;en C es el interior del if(led==L0R || led==L0G || led==L0B)

;function=0x4;

mov r3, #4 ;asigno el valor 4 a r3

strb r3, [fp, #-8] ;fp-8 corresponde a la variable function

;port\_gpio=5;

mov r3, #5

strb r3, [fp, #-6] ;fp-6 corresponde a la variable port\_gpio

;if(led==L0R)

ldrb r3, [fp, #-13] @ zero\_extendqisi2

;led==L0R

cmp r3, #0

bne .L4 ;si es negativa el resultado de la comparación:

;entonces pasa al siguiente “else if”

;pin=0;

mov r3, #0

strb r3, [fp, #-5] ;fp-5 corresponde a la variable pin

;pin\_gpio=0;

mov r3, #0

strb r3, [fp, #-7] ;fp-7 corresponde a la variable pin\_gpio

b .L11 ;sale fuera del if

.L4: ;else if(led==L0G)

;led==L0G

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #1

bne .L6 ;si es negativa el resultado de la comparación:

;entonces pasa al siguiente “else if”

;pin=1;

mov r3, #1

strb r3, [fp, #-5]

;pin\_gpio=1;

mov r3, #1

strb r3, [fp, #-7]

b .L11 ;sale fuera del if

.L6: ;else if(led==L0B)

;led==L0B

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #2

bne .L11 ;si es negativa la comparación sale fuera del if

mov r3, #2

strb r3, [fp, #-5]

mov r3, #2

strb r3, [fp, #-7]

b .L11 ;sale fuera del if

.L3: ;else if(led==L1 || led==L2 || led== L3)

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #3

beq .L8 ;si el resultado de la comparacion es true entro al else if

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #4

beq .L8 ;si el resultado de la comparacion es true entro al else if

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #5

bne .L7 ;salto a .L7 si resultado de cmp es false → fuera del if

.L8: ;en C es el interior de else if(led==L1 || led==L2 || led== L3)

;function=0x0;

mov r3, #0

strb r3, [fp, #-8]

if(led==L1)

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #3

bne .L9 ;salto a .L9 si resultado de cmp es false → sig else if

mov r3, #10

strb r3, [fp, #-5]

mov r3, #0

strb r3, [fp, #-6]

mov r3, #14

strb r3, [fp, #-7]

b .L7 ;salto a .L7 incondicionalmente→ fuera del if

.L9: ;if(led==L2)

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #4

bne .L10 ;salto a .L10 si resultado de cmp es false → sig else if

mov r3, #11

strb r3, [fp, #-5]

mov r3, #1

strb r3, [fp, #-6]

mov r3, #11

strb r3, [fp, #-7]

b .L7 ;salto a .L7 incondicionalmente→ fuera del if

.L10: ;else if(led==L3)

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #5

bne .L7 ;salto a .L7 si resultado de cmp es false → fuera del if

mov r3, #12

strb r3, [fp, #-5]

mov r3, #1

strb r3, [fp, #-6]

mov r3, #12

strb r3, [fp, #-7]

b .L7 ;salto a .L7 incondicionalmente→ fuera del if

.L11:

nop

.L7: ; fuera del if

;SCU\_SetPin(SCU, port, pin, function);

ldrb r3, [fp, #-8] @ zero\_extendqisi2

ldrb r2, [fp, #-5] @ zero\_extendqisi2

ldrb r1, [fp, #-9] @ zero\_extendqisi2 ;carga los valores de function, port y pin

ldr r0, .L12 ;carga la direccion de SCU en r0

bl SCU\_SetPin

;GPIO\_SetPinDIR(GPIO,port\_gpio, pin\_gpio,OUTPUT);

ldrb r2, [fp, #-7] @ zero\_extendqisi2

ldrb r1, [fp, #-6] @ zero\_extendqisi2; ;carga los valores pin\_gpio y port\_gpio

mov r3, #1 ;carga output equivalente a 1

ldr r0, .L12+4;r1 <-- direccion de r1 con un offset de (r2<<2)

;carga la direccion de GPIO

bl GPIO\_SetPinDIR

nop

sub sp, fp, #4

@ sp needed

pop {fp, lr}

bx lr

.L13:

.align 2

.L12:

.word 1074290688 ;Direccion de SCU

.word 1074741248 ;Direccion de GPIO

.size LED\_cfg, .-LED\_cfg

;declaracion funcion LED\_high

.align 2

.global LED\_high

.syntax unified

.arm

.fpu softvfp

.type LED\_high, %function

LED\_high:

@ Function supports interworking.

@ args = 0, pretend = 0, frame = 16

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

push {fp, lr}

add fp, sp, #4

sub sp, sp, #16

mov r3, r0

strb r3, [fp, #-13]

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #0

beq .L15

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #1

beq .L15

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #2

bne .L16

.L15:

mov r3, #5

strb r3, [fp, #-5]

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #0

bne .L17

mov r3, #0

strb r3, [fp, #-6]

b .L24

.L17:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #1

bne .L19

mov r3, #1

strb r3, [fp, #-6]

b .L24

.L19:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #2

bne .L24

mov r3, #2

strb r3, [fp, #-6]

b .L24

.L16:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #3

beq .L21

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #4

beq .L21

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #5

bne .L20

.L21:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #3

bne .L22

mov r3, #0

strb r3, [fp, #-5]

mov r3, #14

strb r3, [fp, #-6]

b .L20

.L22:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #4

bne .L23

mov r3, #1

strb r3, [fp, #-5]

mov r3, #11

strb r3, [fp, #-6]

b .L20

.L23:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #5

bne .L20

mov r3, #1

strb r3, [fp, #-5]

mov r3, #12

strb r3, [fp, #-6]

b .L20

.L24:

nop

.L20:

ldrb r2, [fp, #-6] @ zero\_extendqisi2

ldrb r1, [fp, #-5] @ zero\_extendqisi2

mov r3, #1

ldr r0, .L25

bl GPIO\_SetPin

nop

sub sp, fp, #4

@ sp needed

pop {fp, lr}

bx lr

.L26:

.align 2

.L25:

.word 1074741248

.size LED\_high, .-LED\_high

;declaracion funcion LED\_low

.align 2

.global LED\_low

.syntax unified

.arm

.fpu softvfp

.type LED\_low, %function

LED\_low:

@ Function supports interworking.

@ args = 0, pretend = 0, frame = 16

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

push {fp, lr}

add fp, sp, #4

sub sp, sp, #16

mov r3, r0

strb r3, [fp, #-13]

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #0

beq .L28

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #1

beq .L28

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #2

bne .L29

.L28:

mov r3, #5

strb r3, [fp, #-5]

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #0

bne .L30

mov r3, #0

strb r3, [fp, #-6]

b .L37

.L30:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #1

bne .L32

mov r3, #1

strb r3, [fp, #-6]

b .L37

.L32:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #2

bne .L37

mov r3, #2

strb r3, [fp, #-6]

b .L37

.L29:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #3

beq .L34

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #4

beq .L34

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #5

bne .L33

.L34:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #3

bne .L35

mov r3, #0

strb r3, [fp, #-5]

mov r3, #14

strb r3, [fp, #-6]

b .L33

.L35:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #4

bne .L36

mov r3, #1

strb r3, [fp, #-5]

mov r3, #11

strb r3, [fp, #-6]

b .L33

.L36:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #5

bne .L33

mov r3, #1

strb r3, [fp, #-5]

mov r3, #12

strb r3, [fp, #-6]

b .L33

.L37:

nop

.L33:

ldrb r2, [fp, #-6] @ zero\_extendqisi2

ldrb r1, [fp, #-5] @ zero\_extendqisi2

mov r3, #0

ldr r0, .L38

bl GPIO\_SetPin

nop

sub sp, fp, #4

@ sp needed

pop {fp, lr}

bx lr

.L39:

.align 2

.L38:

.word 1074741248

.size LED\_low, .-LED\_low

;declaracion funcion LED\_toggle

.align 2

.global LED\_toggle

.syntax unified

.arm

.fpu softvfp

.type LED\_toggle, %function

LED\_toggle:

@ Function supports interworking.

@ args = 0, pretend = 0, frame = 16

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

push {fp, lr}

add fp, sp, #4

sub sp, sp, #16

mov r3, r0

strb r3, [fp, #-13]

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #0

beq .L41

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #1

beq .L41

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #2

bne .L42

.L41:

mov r3, #5

strb r3, [fp, #-5]

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #0

bne .L43

mov r3, #0

strb r3, [fp, #-6]

b .L50

.L43:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #1

bne .L45

mov r3, #1

strb r3, [fp, #-6]

b .L50

.L45:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #2

bne .L50

mov r3, #2

strb r3, [fp, #-6]

b .L50

.L42:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #3

beq .L47

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #4

beq .L47

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #5

bne .L46

.L47:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #3

bne .L48

mov r3, #0

strb r3, [fp, #-5]

mov r3, #14

strb r3, [fp, #-6]

b .L46

.L48:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #4

bne .L49

mov r3, #1

strb r3, [fp, #-5]

mov r3, #11

strb r3, [fp, #-6]

b .L46

.L49:

ldrb r3, [fp, #-13] @ zero\_extendqisi2

cmp r3, #5

bne .L46

mov r3, #1

strb r3, [fp, #-5]

mov r3, #12

strb r3, [fp, #-6]

b .L46

.L50:

nop

.L46:

ldrb r2, [fp, #-6] @ zero\_extendqisi2

ldrb r3, [fp, #-5] @ zero\_extendqisi2

mov r1, r3

ldr r0, .L51

bl GPIO\_TogglePin

nop

sub sp, fp, #4

@ sp needed

pop {fp, lr}

bx lr

.L52:

.align 2

.L51:

.word 1074741248

.size LED\_toggle, .-LED\_toggle

;declaración funcion LED\_ALL

.align 2

.global LED\_ALL

.syntax unified

.arm

.fpu softvfp

.type LED\_ALL, %function

LED\_ALL:

@ Function supports interworking.

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

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

push {fp, lr}

add fp, sp, #4

mov r0, #3

bl LED\_cfg

mov r0, #4

bl LED\_cfg

mov r0, #5

bl LED\_cfg

mov r0, #0

bl LED\_cfg

mov r0, #1

bl LED\_cfg

mov r0, #2

bl LED\_cfg

nop

sub sp, fp, #4

@ sp needed

pop {fp, lr}

bx lr

.size LED\_ALL, .-LED\_ALL

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