

## gpio.s Source Code: Kyler Fillerup & Casey Nordgran – Group Partnership #32

```
.equ GPIO_OUT_BASE, 0x40013000
.equ GPIO_OUT, 0x40013088
    .text
    .syntax unified
    .thumb

@ Configure the GPIOx to output
@ Inputs: GPIO number is provided in r0
@ Output:
    .global initGPIO
    .type    initGPIO, %function
initGPIO:
    @ Load GPIO_OUT_BASE address
        movw  r1, #:lower16:GPIO_OUT_BASE
        movt  r1, #:upper16:GPIO_OUT_BASE
    @ Calculate the GPIOx register address
        lsl   r0, 2
        add   r1, r0

    @ Write 1 to config register to set GPIO as output
        mov   r2, #0x1
        str   r2, [r1, #0]
        .rept 7 @ repeat block increments address by 4,
            add   r1, #4 @ and configs next GPIO to output
            str   r2, [r1, #0]
        .endr
        bx    lr @ Return

@ Set the value of all 32 GPIO output bits based on the input bits
@ Inputs: 32bit value is provided in r0
@ Output:
    .global setGPIO
    .type    setGPIO, %function
setGPIO:
    @ Load GPIO_OUT register address
        movw  r1, #:lower16:GPIO_OUT
        movt  r1, #:upper16:GPIO_OUT
    @ Write 32bit value to GPIO output register
        str   r0, [r1, #0]

        mov   r2, #1000
delayloop:                                @ nested loops to add delay
        mov   r3, #1000
delayloop2:
        subs  r3, #1
        bne   delayloop2
        subs  r2, #1
        bne   delayloop
        bx    lr @return
```