

**Library Components 2**

Bryan Satterfield & Jason Maurer

**Multi-Read & Multi-Write**

| **Void E4235\_MultiGPIOwrite(char [] \* pins, int value)** |
| --- |
| **Overview:**  This function sets the output state of multiple specified pins to either HIGH or LOW. |
| **Parameters:**  char [] \* pins - The pins are entered as a string to be parsed to select which GPIOs to write to.  value - The output state to be written to the GPIO pin. The value is either HIGH or LOW. HIGH(1) setting the output to HIGH and LOW(0) to LOW. |
| **Returns:**  none |

**Description:**

This program sets the function of the desired GPIO pins to write either High or LOW. High or Low being predefined constants that are 1 or 0 respectively.

**Example C:**

Int main ()

{

4235\_GPIOwrite(“1 2 3 4-6”, HIGH);

}

**Example Assembly:**

LDR r0, input

bl E4235\_GPIOMultiread

.data

Input: .asciz “1 2 3”

| **Int E4235\_GPIOMultiread (char [] \* , int value)** |
| --- |
| **Overview:**  This function sets the input state of the specified pin to read for either LOW or HIGH. |
| **Parameters:**  char [] \* pins - The pins are entered as a string to be parsed to select which GPIOs to set to read.  value - The input state for the GPIO pin to read for. The value is either HIGH or LOW. HIGH(1) setting the pin to read for HIGH and LOW(0) reading for LOW. |
| **Returns:**  Positive Integer number that represents which pins are high. |

**Description:**

This program sets the desired GPIO pins to READ. It then returns the state read by each GPIO pin in the form of a positive integer number. This integer number can then be compared to a binary string such as 0b111, 0b110, or 0b000 where each bit represents the pin. In the order they were entered. If the GPIO pin is set to read HIGH then it will return 1 if a high signal is read or 0 if low signal. If the GPIO is set to read LOW then it will return 1 if a low signal is read or 0 if a high signal is read.

**Example C:**

Int main()

{

4235\_GPIOMultiread(“1-3”, HIGH);

Int output = 4235\_GPIOMultiread(20,HIGH);

if(output == 0b111)

printf(“all pins are high)

}

**Example Assembly:**

Ldr r0, input

bl E4235\_GPIOMultiread

cmp r0, 0b111

beq fun

.

.

.

.data

Input: .asciz “1 2 3”

**Test Plan**

**E4235\_GPIOMultiwrite():**

Connect three GPIO pins to a resistor and Diode. Then call E4235\_GPIOMultiwrite on those three pins. First to write HIGH to all of them then check to see if they are on. Then write LOW to all of them and check if they all turn off. Then move to another set of three pins and check several combinations till sufficient. With different calls such as “1 2 3” or “1-3”. Repeat this process for several pins.

**E4235\_GPIOMultiread():**

Connect three GPIO pins to three other pins that are either high or low. Then after reading the three pins print out the returned binary string representing the states of the three read pins e.g. (111,101,110 … 000). Then move to another set of three pins and check several combinations till sufficient. With different calls such as “1 2 3” or “1-3”. Repeat this process for several pins.