

**Library Components 1**

Bryan Satterfield & Jason Maurer

**Read & Write**

| **Void E4235\_GPIOwrite(int GPIOnumber, int value)** |
| --- |
| **Overview:**  This function sets the output state of the specified pin to either HIGH or LOW. |
| **Parameters:**  GPIOnumber - The GPIOnumber of the GPIO pin to be set for outputting a value. This number must be in between 1-27. For each corresponding GPIO pin.  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 pin to write either High or LOW. High or Low being predefined constants that are 1 or 0 respectively.

**Example C:**

Int main ()

{

4235\_GPIOwrite(21, HIGH);

}

**Example Assembly:**

mov R0, #21

mov R1, #1

bl gPIOwrite

| **Int E4235\_GPIOread (int GPIOnumber, int value)** |
| --- |
| **Overview:**  This function sets the input state of the specified pin to read for either LOW or HIGH. |
| **Parameters:**  GPIOnumber - The GPIO number of the GPIO pin to be set for an input. This number must be in between 1-27. For each corresponding GPIO pin.  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:**  Int 1 or Int 0 |

**Description:**

This program sets the desired GPIO pin to READ. It then returns the state read by the GPIO pin. 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\_GPIOwrite(21, HIGH);

If (4235\_GPIOread(20,HIGH))

printf(“pin is high”)

}

**Example Assembly:**

mov R0, #20

mov R1, #1

bl gPIOread

**Select**

| **Void E4235\_GPIOselect (int GPIOnumber, int function)** |
| --- |
| **Overview**  This function sets the function of the specified pin to be either READ or WRITE. |
| **Parameters:**  GPIOnumber - The GPIO number of the GPIOpin to be set for a function. This number must be in between 1-27. For each corresponding GPIO pin.  function - The function for the pin to be set to. To set a pin to be a ‘Write’ pin, input ‘0’ or ‘WRITE’. To set a pin to be a ‘Read’ pin, input ‘1’ or ‘READ’. |
| **Returns:**  none |

**Description:**

This function sets the GPIO function of the pin to either Write or Read. This is performed by calling the E4235\_GPIOread or E4235\_GPIOwrite These then load corresponding register with the correct mask to the desired pin to either read or write

**Example C:**

Int main()

{

4235\_GPIOselect(21, WRITE);

}

**Example Assembly:**

mov R0, #20

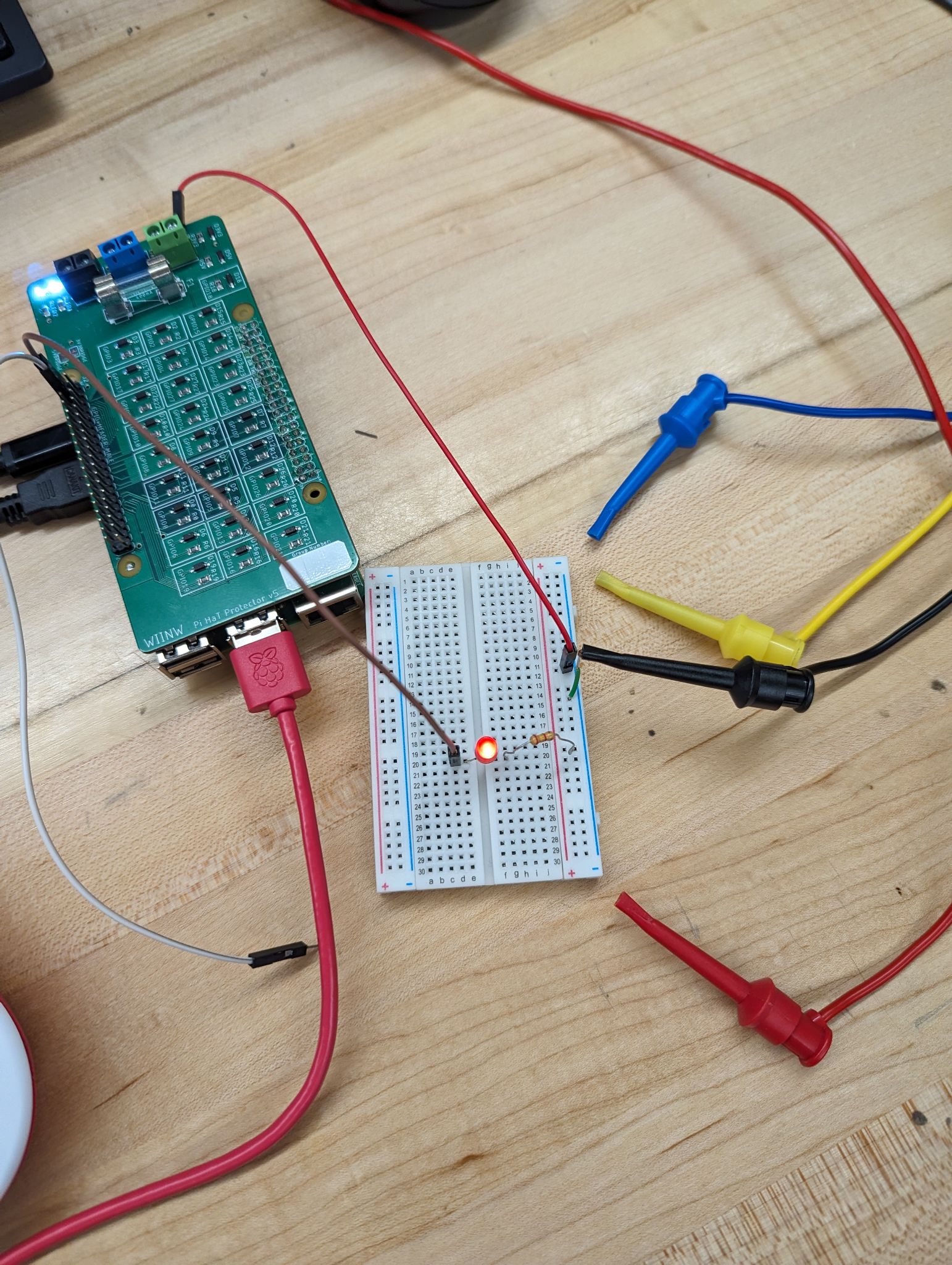
mov R1, #0

bl gPIOselect

**Test Plan**

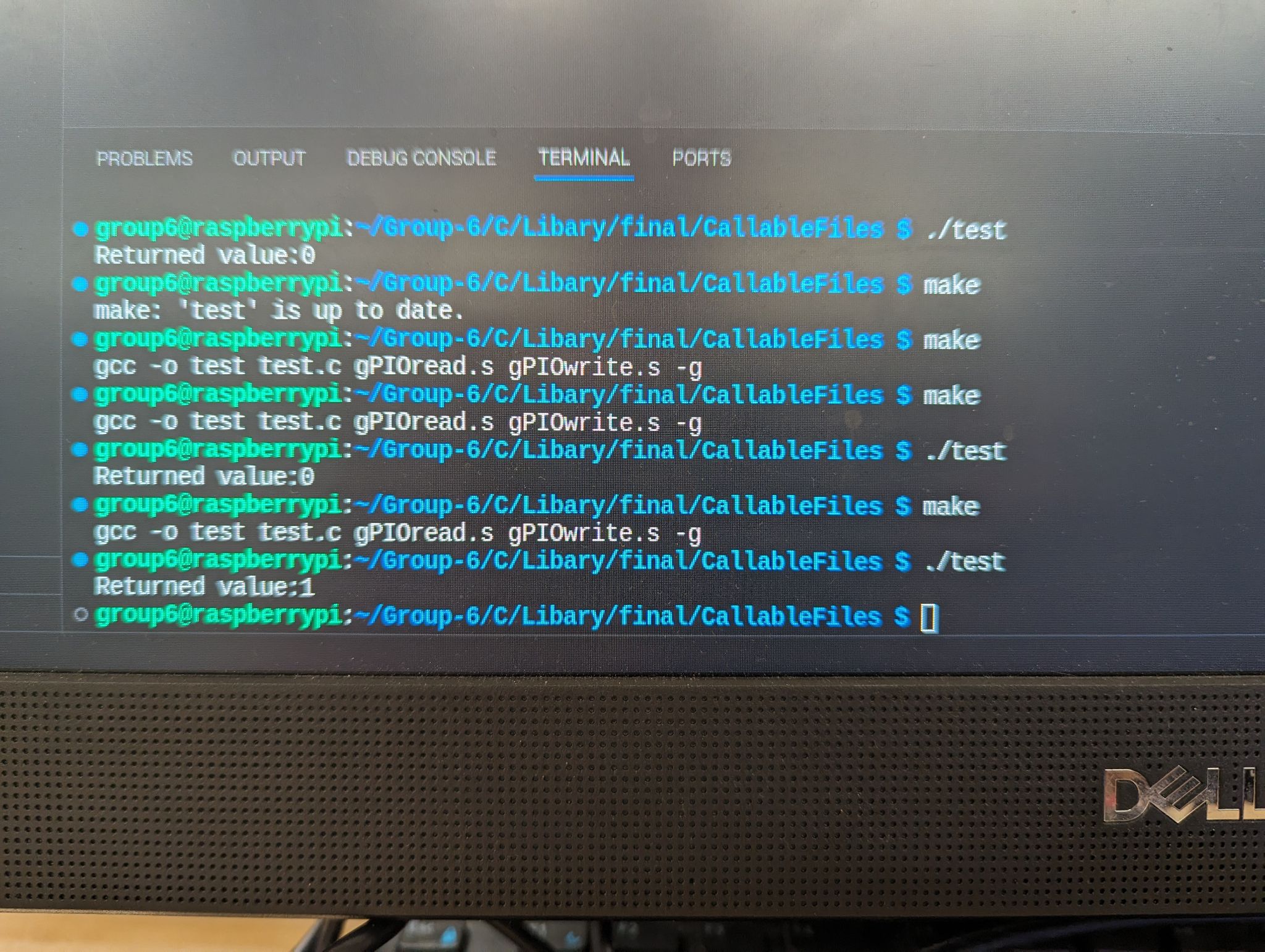
**E4235\_GPIOwrite():**

Connect a GPIO pin to a diode and resistor circuit. First write HIGH to the pin and see if the diode lights ups. If so then the function has successfully changed the output of the pin to HIGH. With the same pin still connected to the circuit write the output to be LOW and see if the diode turns off. If so then the function has successfully changed the output of the pin to be LOW. Repeat this process for all GPIO pins to ensure the function can change the output for all GPIO pins.



**E4235\_GPIOread():**

With the write function now working we can now attach two pins together using a wire. Then set one GPIO pin to output HIGH or LOW with write then attach it to every other GPIO pin one at a time. With each pin it is attached to, use read on that attached pin to read the state of the write pin. Using this we can then output the state of the write pin to the terminal checking if it matches the state of the pin that is being read. If it does match then the read function works for that pin. Repeat this process using one pin as the write pin then attached to all other pins for them to read it.



**E4235\_GPIOselect():**

Now that both functions work we can then use select to set the function of two pins. One the outputs and one that reads that output as input. Using a similar process used for testing read. We can test several pairs of GPIO pins. With one as output and one reading that output and then displaying to the terminal. Then switching the direction by changing the write pin to read and the read pin to write.