Phillip Raspe

Problem3

(a)

virtual int setDirection(GPIO\_DIRECTION); //header call for setDirection, takes a direction type as a parameter and returns a virtual int.

int GPIO::setDirection(GPIO\_DIRECTION dir) //cpp implementation of setDirection

{

switch(dir) //a switch statement to identify what “dir” is

{

case INPUT: //in the case of INPUT

return this->write(this->path, "direction", "in"); //write “in” in the “direction” field at “this path”

break; // break after this case

case OUTPUT: //in the case of OUTPUT

return this->write(this->path, "direction", "out"); //write “out” in the “direction” field at “this path”

break; // break after this case

}

return -1; //return -1

}

(b)

virtual GPIO\_VALUE getValue(); //header call for getValue takes no parameters and returns a virtual type

GPIO\_VALUE GPIO::getValue() //cpp implementation of getValue

{

string input = this->read(this->path, "value"); // reads the data stored in the “value” field at “this path” and puts it in a string called “input”

if (input == "0") // if the value of input is a “0”

return LOW; //return virtual LOW

else //otherwise

return HIGH; //return virtual HIGH

}