getTemp.c Alexander Dean

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <linux/i2c-dev.h>
#include <string.h>

int main() {
    unsigned int temp = 0;
    int errChk;
    int i2c_fd;
    char *buffer;
    char *cmd;
    char *strcmd;
    cmd = malloc(100);
    if (cmd == NULL) {
        perror("MALLOC");
        return 1;
    }
    buffer = malloc(10);
    if (buffer == NULL) {
        perror("MALLOC");
        return 1;
    }

    if ((i2c_fd = open("/dev/i2c-1", O_RDWR)) < 0) {        //OPENS I2C DEVICE TO READ AND WRI
TE
        perror("I2C OPEN");
        return 1;
    }
    if (ioctl(i2c_fd, I2C_SLAVE, 0x48) < 0) {           //SETS I2C SLAVE ADDRESS TO 0x48, THE
DESIGNATED ADDRESS FOR THE TC74A0
        perror("I2C IOCTL");
        return 1;
    }
    if ((write(i2c_fd, buffer, 1)) != 1) {              //SELECTS REGISTER 0 AS REGISTER TO RE
AD FROM
        perror("I2C WRITE");
        return 1;
    }
    if (read(i2c_fd, buffer, 1) != 1) {             //GETS A BYTE OF DATA FROM I2C DATA LINE
        perror("I2C READ");
        return 1;
    }
    strcpy(cmd, "php -f /home/pi/Documents/ECE331/templogger/storeTemp.php temp=");     //PREP
ARES COMMAND LINE COMMAND
    buffer[0] = ((buffer[0] * 9) / 5) + 32;                         //CONVERTS TEMPERATURES TO
 FARENHEIT BECAUSE WE'RE AMERICAN!
    sprintf(buffer, "%d", buffer[0]);                           //PERFORMS INTEGER TO STRING C
ONVERSION ON TEMPERATURE VALUE
    strcat(cmd, buffer);                                    //COMBINES TEMPERATURE WITH COMMAN
D LINE ARGUMENT
    errChk = system(cmd);                                   //CALLS PASSED COMMAND
    if (errChk != 0) {
        perror("SYSTEM");
        return 1;
    }
    return 0;
}
```