LOCK GUIDE

Last Update: Dec 3 2017

This document guides the user through:

1. Soldering and powering for 5V relay
2. Connecting and powering a lock.
3. Controlling lock through the BeagleBone via command line
4. Using a C program to control lock

Table of Contents

Formatting:

1. Host (desktop) commands starting with $ are Linux console commands
   $ echo “Hello world”
2. Target (board) commands start with #:
   # echo “On embedded board”
3. Almost all commands are case sensitive.

Permission of copyright:

1. MATERIALS

Before listing out the materials we suggest you follow the GPIO guide on the CMPT433 website:

http://www.cs.sfu.ca/CourseCentral/433/bfraser/other/GPIOGuide.pdf - By Dr. Brain Fraser

The materials for this guide is as follows:

1. Jumper cables - Male-to-Female (3x)
2. Jumper cables – Male-to-Male (2x)
3. One relay control kit (The one used in this guide is purchased from the following link):
   https://www.sparkfun.com/products/13815
4. One Lock-style Solenoid (The one used in this guide was purchased from the following link):
   https://www.adafruit.com/product/1512
5. 12V DC power source (The one used in this guide was NOT purchased from the following link, but this is a similar item and serves the same purpose):
   https://www.ebay.com/itm/AC-DC-100-240V-12V-1A-Power-Supply-Adapter-Charger-For-3528-5050-LED-Strips-US-/391474239724
6. A Multimeter
7. Wire cutters
8. Soldering Iron
9. Solder (one used in this guide is a “lead free silver solder”)

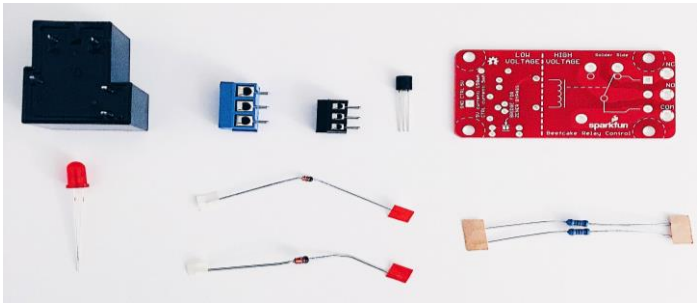
2. SOLDERING PROCESS

NOTE: soldering will not be taught here on this guide it is simply the step by step process on what to solder, but the following link may prove useful:

- https://www.youtube.com/watch?v=Qps9woUGkvI

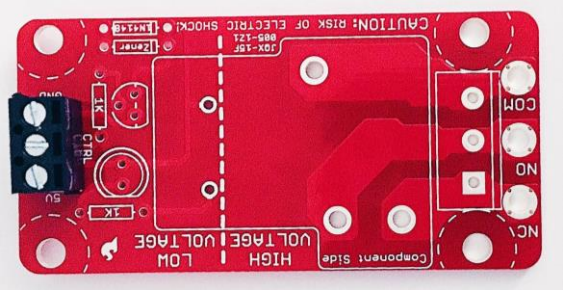
Begin with the relay control kit:

You should have the components from the kit as shown below:

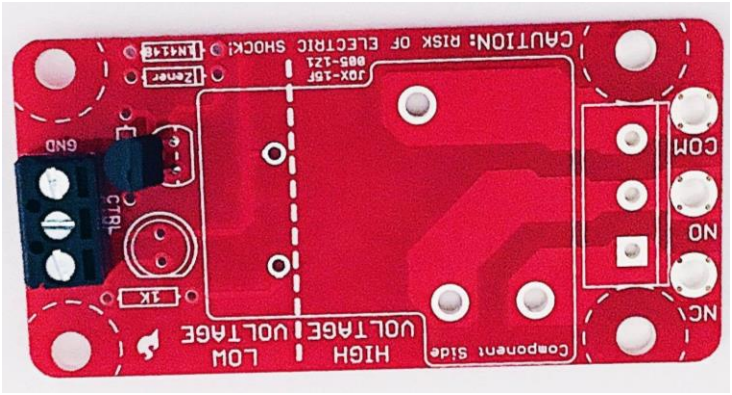


Solder the parts above in the following order (**READ ALL STEPS BEFORE SOLDERING!!**):

1. Solder on the Black screw terminal as such:

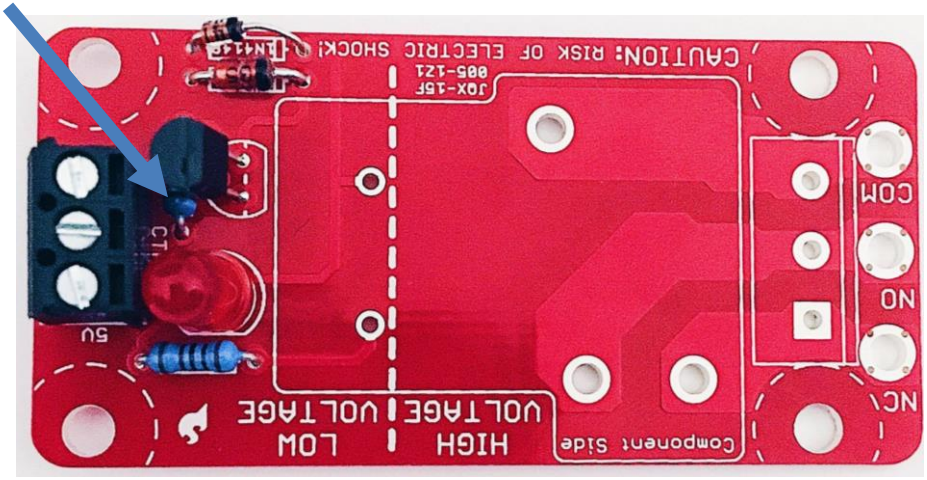


2. Solder on the transistor as such (hint: do step three before this step to make your life easier):

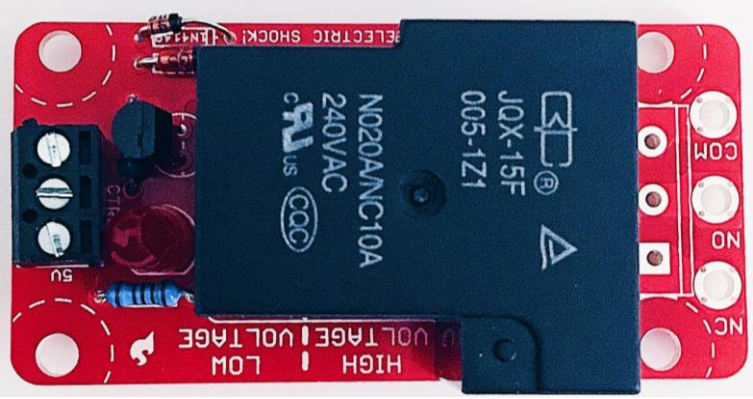


3. Next solder on the diodes and resistors and the led:

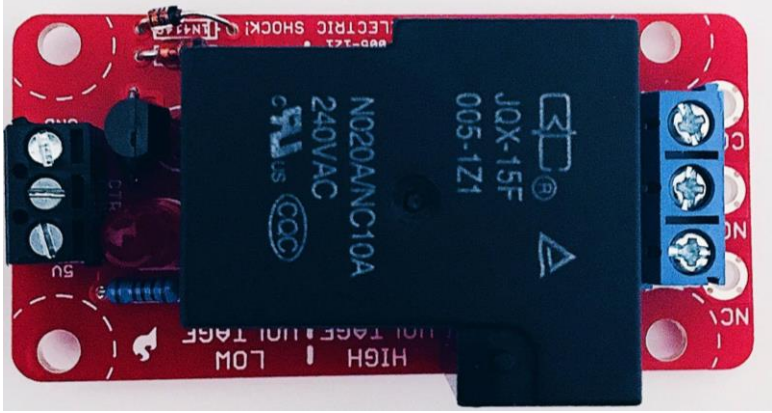
Note the resistor here



4. Next solder on the relay (note the relay has thicker pins for soldering so may need some more time to be heated to begin melting):



5. Finally solder on the remaining screw terminal:



3. POWERING THE RELAY

Now that you have a soldered relay on the circuit board its time to connect it to the BeagleBone and power the relay.

The BeagleBone is going to power this relay through the use of jumper cables (for this section we need Male-to-Female jumper cables(3x))

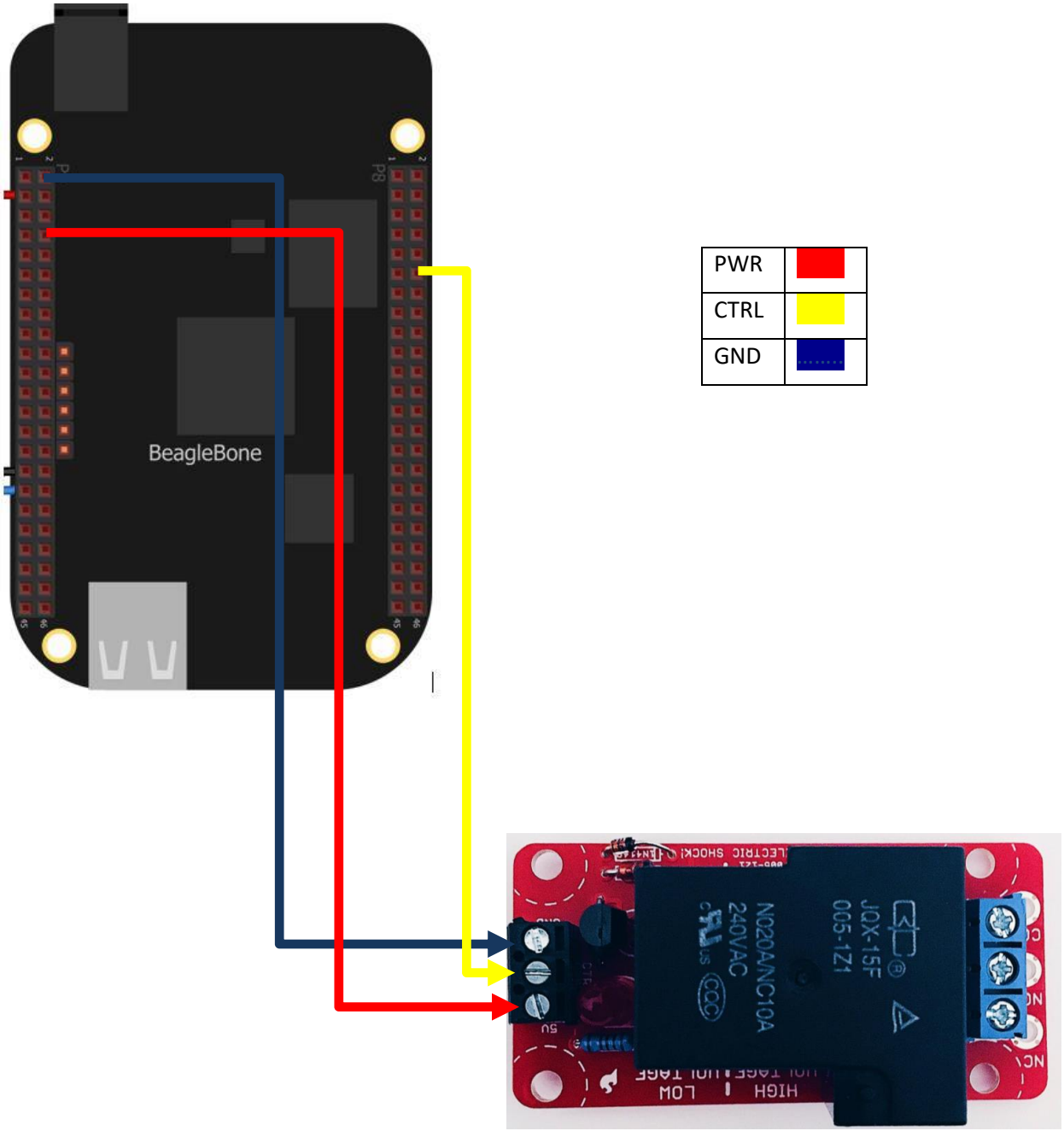
There are pins on the beagle bone to output a constant 5 volts to power the relay called the 5V_SYS output pins, concurrently they are pins P9_7 and P9_8.

Steps (in order):

- Connect one jumper cable from P9_2 to Ground

- Connect one jumper cable from P9_8 to Power
- Connect one jumper cable from P8_12 to Control

The jumper cables will be attached as follows (Thick arrows indicate the male end of the jumper cable)



4. TESTING RELAY

At this point you will have a Relay that you can turn on an off, but of course nothing to turn on and off yet. Nonetheless we can make sure the progress you've made so far is correct by the following test.

SSH into you beaglebone board and cd into your gpio folder:

 # cd /sys/class/gpio

Then export the control pin for which the linux number is 45:

 # echo 45 > export

Run the ls command to make sure you have new directory 'gpio45'

 # ls

cd into the gpio45 folder

 # cd gpio45

Now you can control the relay ON(1) and OFF(0) status through the 'value' parameter

 # echo 0 > value

OR

 # echo 1 > value

You should notice the led on the relay turning on and off and are done for this section.

Run this command to turn off the relay before you proceed

 # echo 0 > value

5. CONNECTING THE LOCK

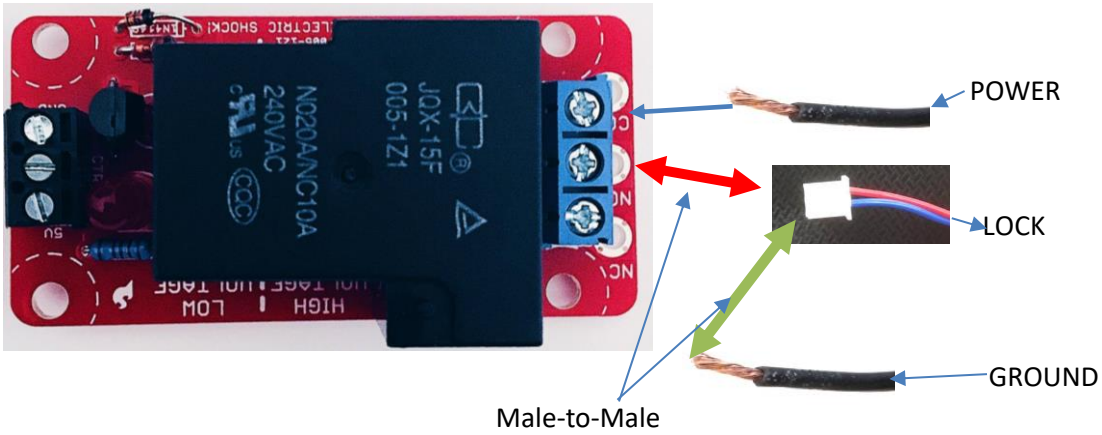
To supply power to the lock you will first need to cut the end of the 12VDC adapter using wire cutters and expose two different wires will be copper.

The next step is to identify which wire supplies power and which wire acts as the ground using the multimeter

The following link will be useful in solving this problem:

- https://www.youtube.com/watch?v=E3elzYUfoe4

Once you have figured out which copper wire is which you can do the following:



Steps:

- Insert the power end into the screw terminal as shown above(shown in blue arrow).
- The next step is to have a Male-to-Male connect to the middle port of the screw terminal to the red port in the solenoid lock(shown in red).
- Have a Male-to-Male connect to the blue port of the solenoid lock(shown in green)
- Then wrap the ground copper wire from the adapter to the other end of the Male-to-Male jumper cable. (Note you can solder this to make it easier to stick)
- Plug in the adapter into a wall for power

6. CONTROLLING THE LOCK

Refer to section 4

7. C CODE

Use the following code to initialize the gpio pin.

```c
void gpioPinInit()
{
    char buff[BUFF_SIZE];

    //add gpio pin to export file
    FILE* pExportFile = fopen("/sys/class/gpio/export","w");
    if(!pExportFile)
    {
        perror("ERROR: ");
        exit(1);
    }
    fprintf(pExportFile, "%d", 45);
    fclose(pExportFile);

    //set direction of pin based on direction parameter
    sprintf(buff,"/sys/class/gpio/gpio%d/direction", 45);
    FILE* pDirectionFile = fopen(buff, "w");
    if(!pDirectionFile)
    {
        perror("ERROR: ");
        exit(1);
    }
    fprintf(pDirectionFile, "%s", "out");
    fclose(pDirectionFile);
}
```

The following code writes to the gpio pin you initialized

```c
void writeToFile(int value)
{
    char buff[BUFF_SIZE];
    sprintf(buff,"/sys/class/gpio/gpio%d/value", 45);
    FILE* pfile = fopen(buff, "w");
    if(!pfile)
    {
        printf("ERROR: Unable to open file! LINE 70\n");
        exit(1);
    }

    if(value == 1 || value == 0)
        fprintf(pfile, "%d", value);
    else
        printf("Expecting only a 1 or a 0 in value\n");
    fclose(pfile);
}
```

Use the following code snippet to run your program

```c
int main()
{
    gpioPinInit();

    while(1)
    {
        writeToFile(1); // turns relay on
        sleep(2) // sleeps for 2 seconds
        writeToFile(0); // turns relay off
    }

    return 1;
}
```