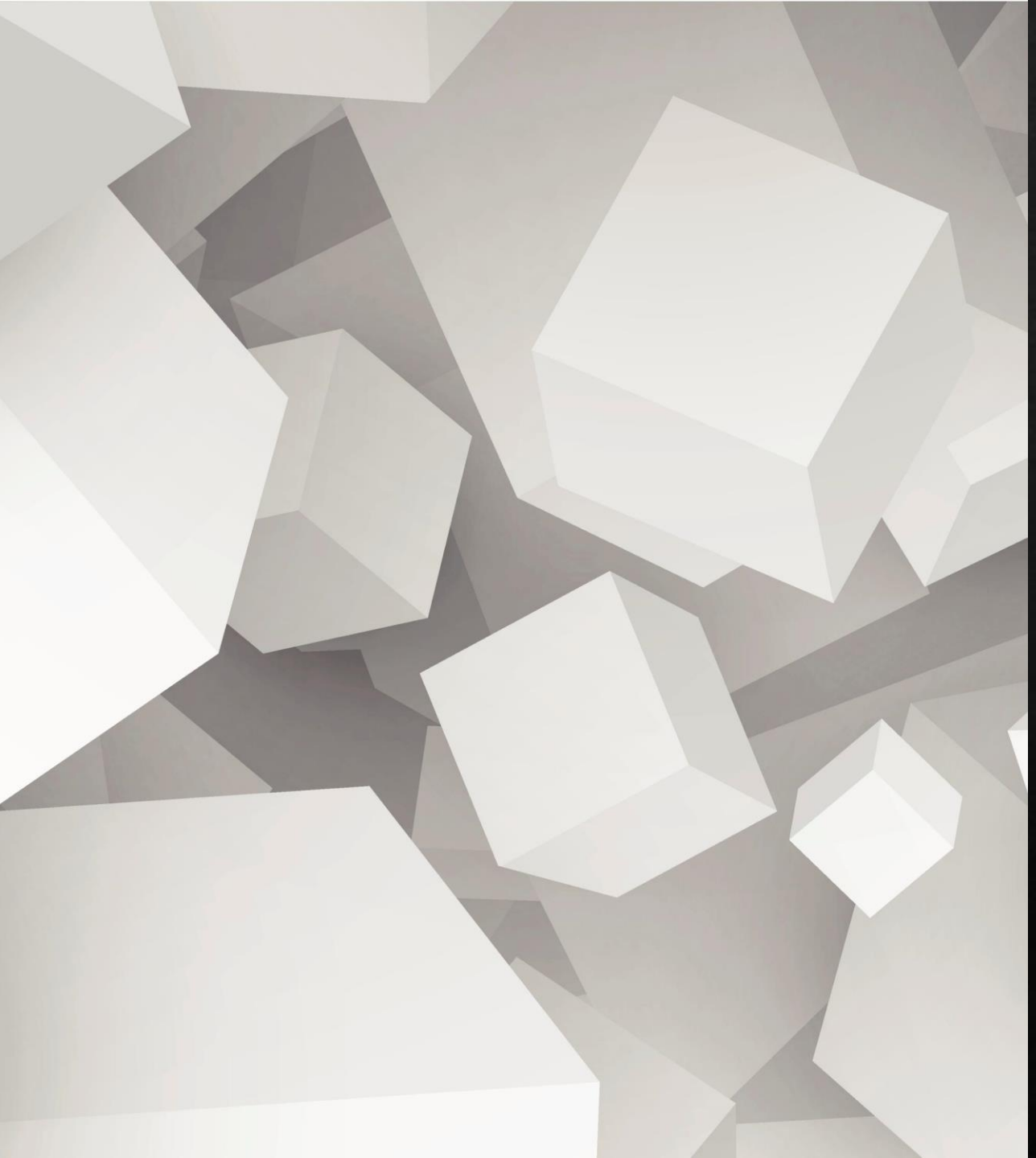
Interfacing with BeagleBone Wireless : Part I

- Zain Rajani (c0752681)

Group # 8



Outline

- Task Introduction
- Tools Requirement
- Interface Programming
- Wire Interface
- Conclusion
- References

Task Introduction

- In the previous meeting we completed the part of designing the schematics i.e. the way one can connect the sensors and other peripherals to the circuit.
- Keeping that schematic in mind one can start the connection to the Beaglebone wireless which is the main processing unit for the project.
- During this task since we have other sensor to connect as well. We have broken the interfacing into parts. Each interfacing will be done simultaneously.
- In the first phase of the project interfacing we will interface the IR sensor and the LED (Light Emitting Diode) to the Beaglebone Wireless.
- The reason for choosing these interfacing in pair is because when the letter arrives in the letter box it will be sensed by the IR sensor and even if a single letter arrives the LED attached to this box will be on (in other words be HIGH)

Task Objective

- Establish a suitable connection between the given sensor, LED and Beaglebone Black Wireless
- The interfacing must be able to count the number of letters in the box and make a LED go on even if a single letter is present in the letterbox.

Task Requirement: Hardware

- In order to accomplish this task we require the following:
  - Beaglebone Black Wireless
  - IR (Infrared) Sensor
  - LED (any color)
  - Connecting Wires
  - Wall Adapter
  - Micro USB Cable*

** For power management purposes only*

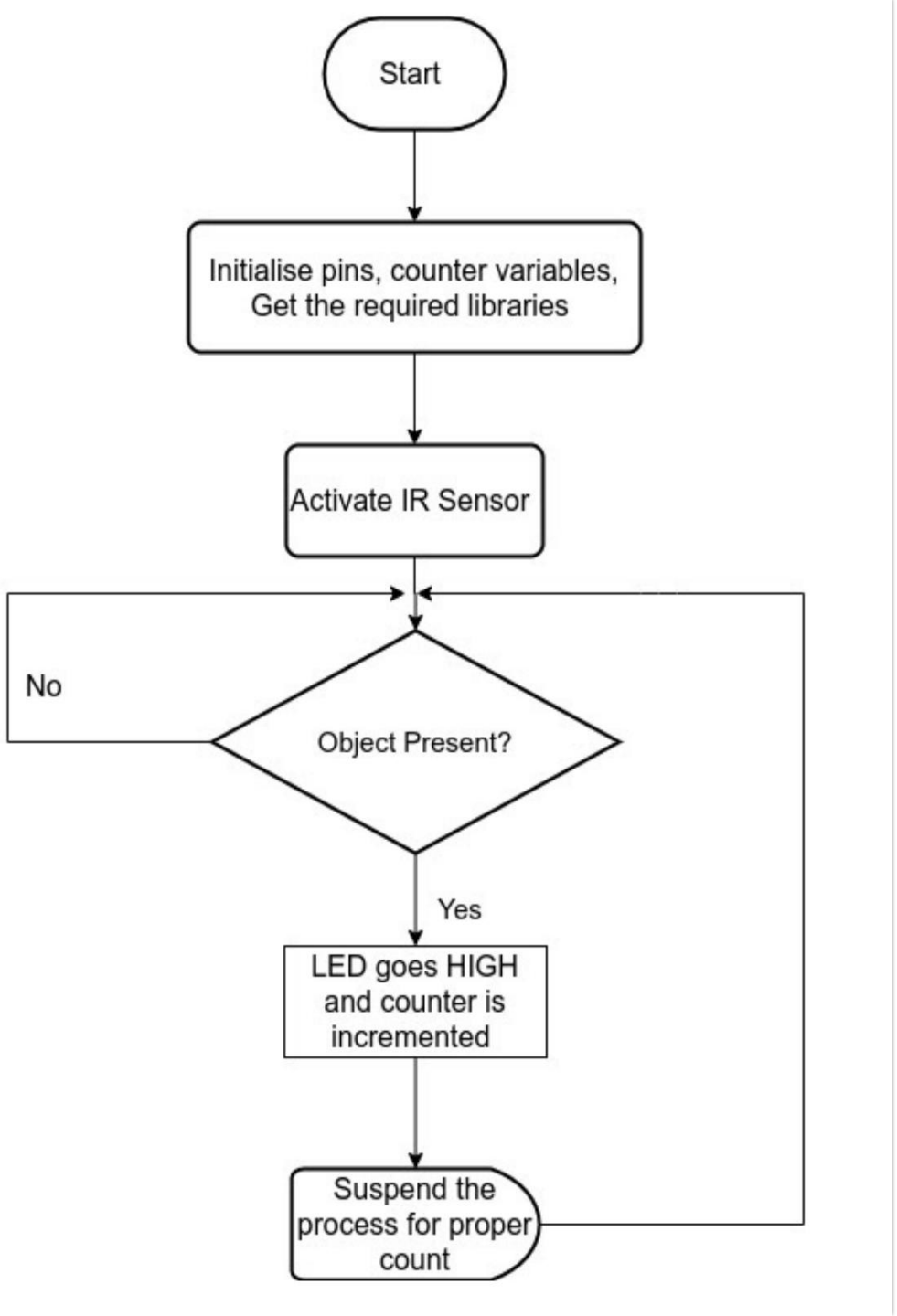
Task Requirement: Software

- Since we must interface the sensor to the Beaglebone Black Wireless (BB-WI) we must program the pins as required for complete fulfillment of the task.
- In our proposal we had already mentioned we shall use Eclipse if the code cannot be written on the Beaglebone directly; but since the interfacing is simple and can be done directly on BB-WI hence we shall use the nano editor from the BB-WI
- Thus to fulfill this requirement one must have the following software present:
  - Debian OS (Installed on the BB-WI)
  - GCC (GNU Not Unix Compiler Collection) to run the code
  - Nano Editor (To write the code)
  - Any required libraries for making interfacing simpler

Interface Programming

- Programming the BB-WI can be very simple as many users over the time has created various headers and function to access the GPIO very easily and hence code them using their function.
- One can choose to write the functions and code the GPIO using the file management concepts in C/C++ or use the inbuilt user defined (created) header file function.
- In our case we downloaded a user defined header file to access the GPIO pins of the BB-WI and programmed them, as necessary. We downloaded a file from the github having the header file name as iobb.h and install it in the BB-WI (*For more refer the reference section)
- The header file had many inbult function avaliable direct to use one just needed to read and study the header file properly

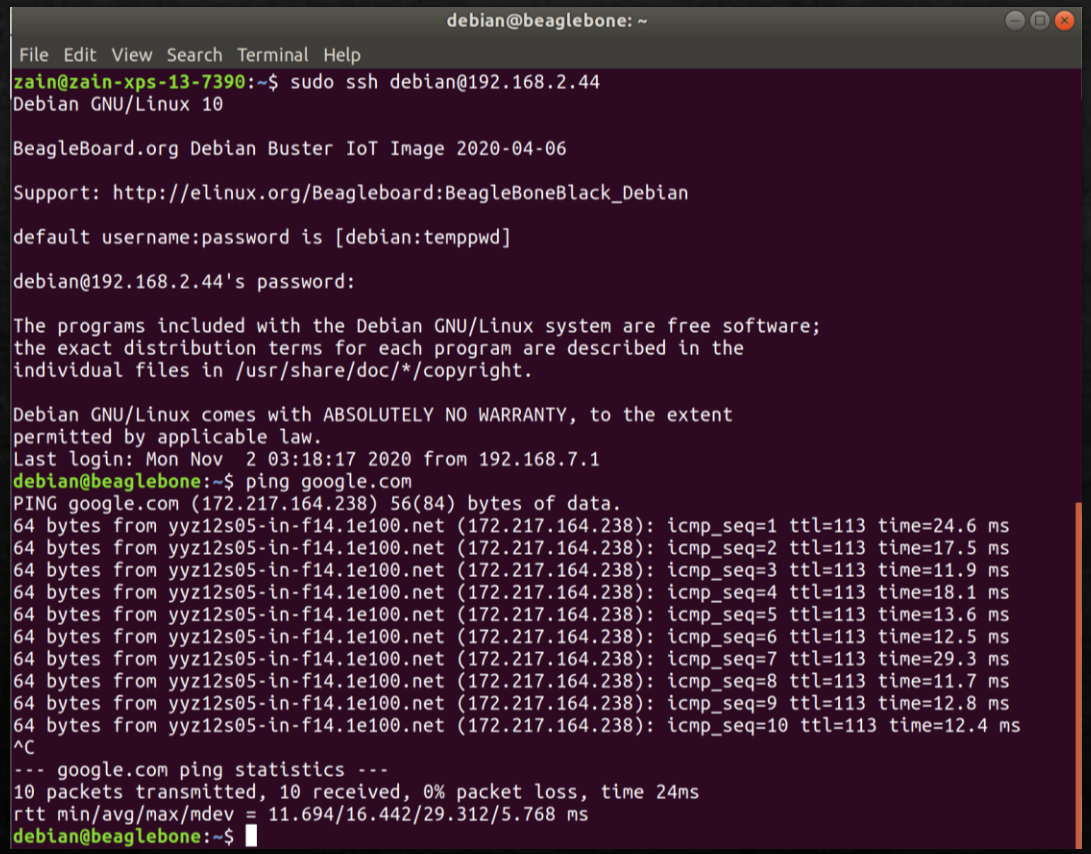
Interface Programming: Flow Daigram

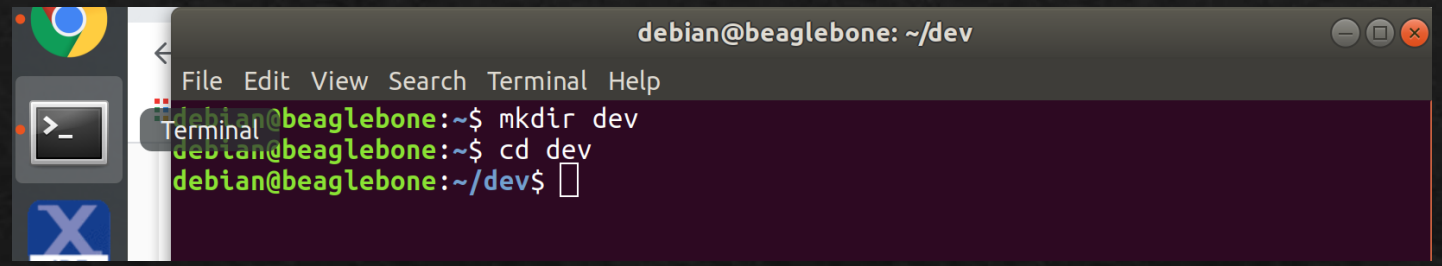


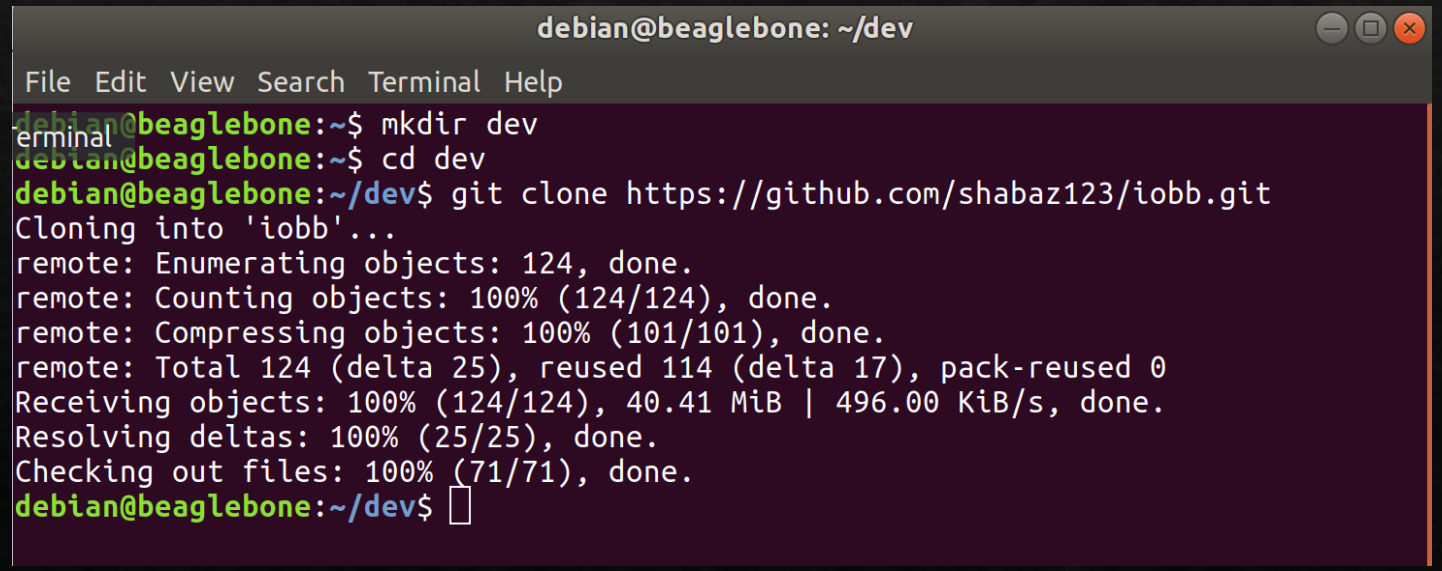
Interface Programming: Library Download

- As mentioned earlier we downloaded the library files from the internet on the Beaglebone Wireless. Here we first need to connect to a valid WiFi or Internet service to download the file.
- We use some common git commands to download this like git clone to get the repository on the BB-WI. Here the difference being that if we download we cannot contribute to the repository and when clone when signficant changes have been made one can ask the original owner to pull those changes for others benefit.
- Thus we first clone the repository, make the file and then make install commands helps for successful installation of the library. These steps can be seen in the following slides

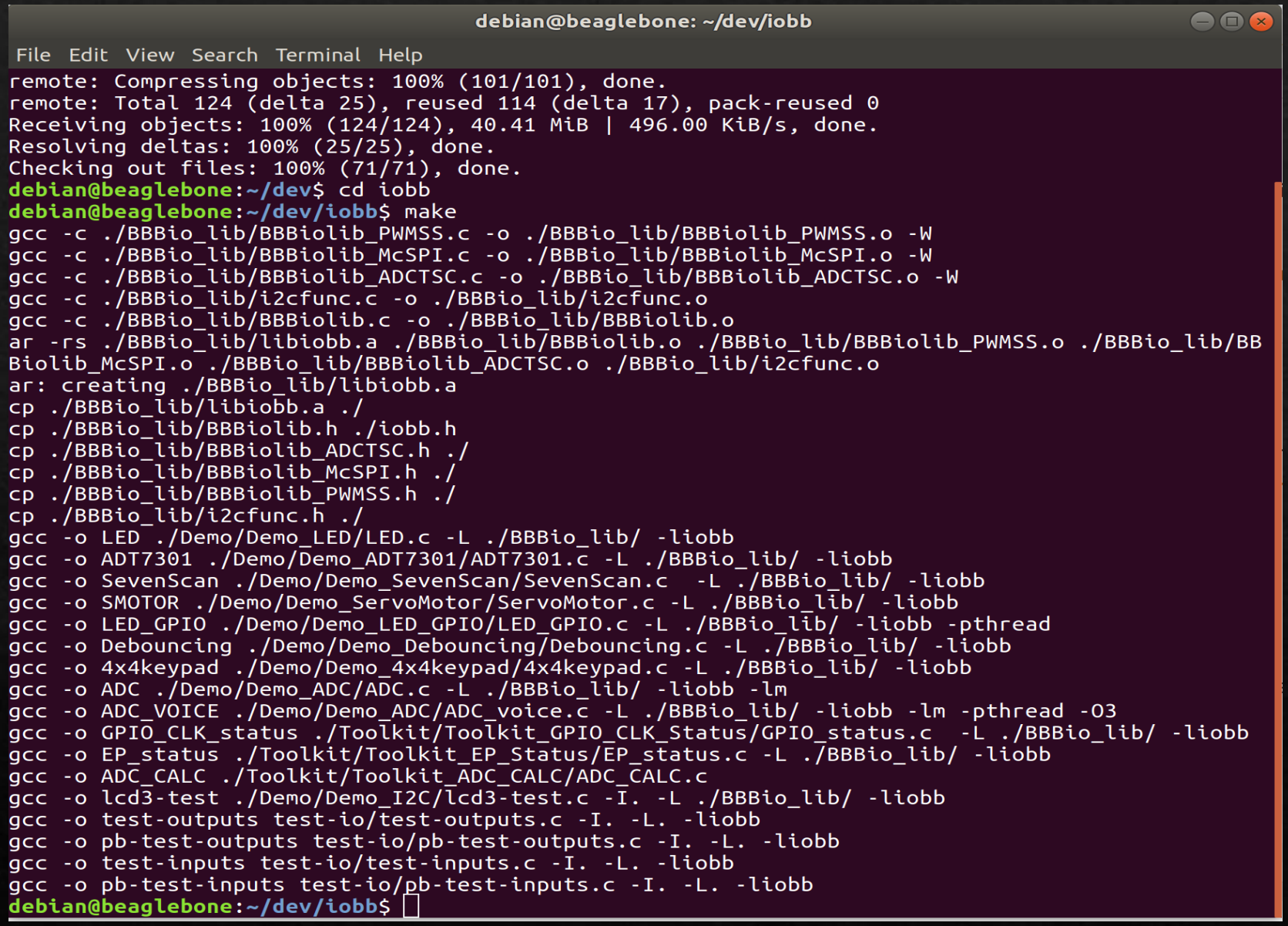
Interface Programming: Library Download



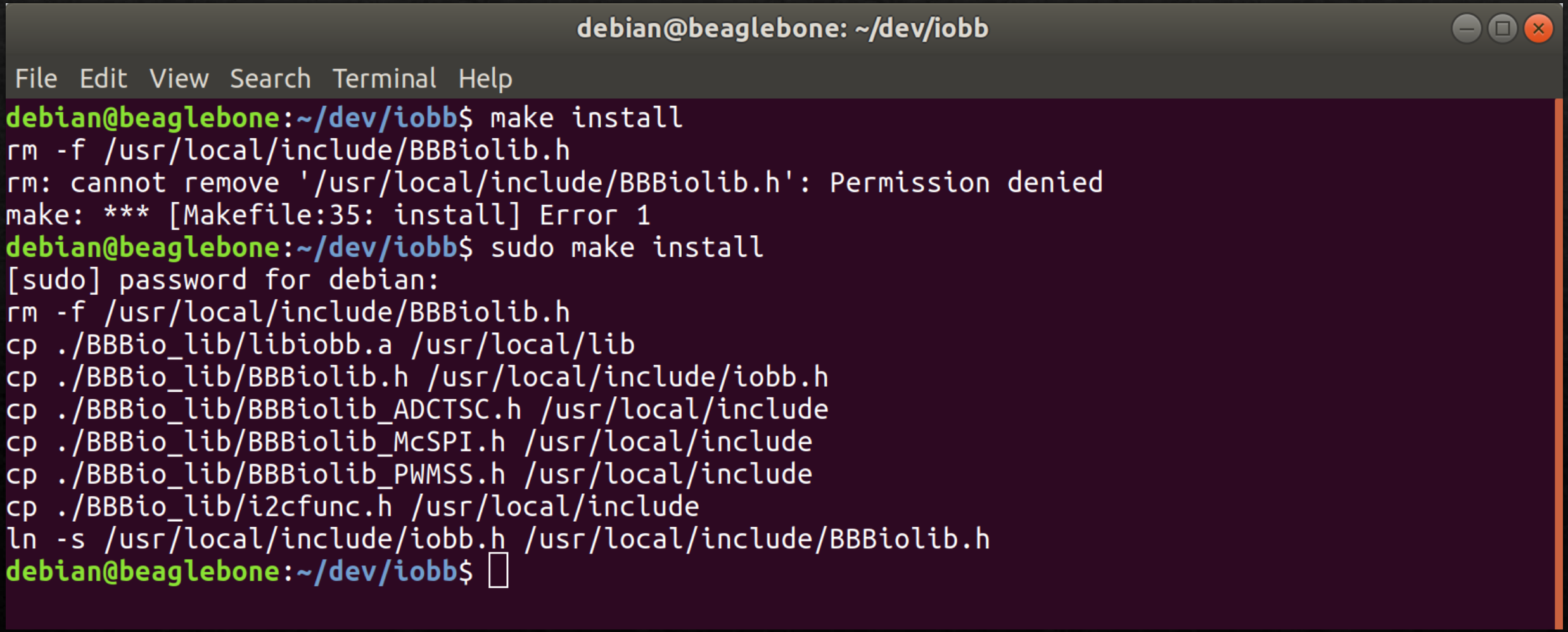




Interface Programming: Library Download



Interface Programming: Library Download



Interface Programming: Coding

- Since we have a flow daigram ready with us one can easily follow it and make the coding task simpler for us as in this course we designed the logic of the code now one just need to think on how can the code become more efficient. This daigram is sometimes also referred to as the algorithm chart or pseudo code.
- First, we import all the libraries which one may require for the programming the interface for example the iobb.h library for accessing the GPIO pins, unistd.h library to get access to some of the POSIX API (this library is already present no requirement to install), stdin.h library to perform the various input/output operation
- Once all the required liberaries are ready one can start to write appropriate code to interface the LED and IR Sensor. For the coding one will first start with the interface the code for IR sensor and then modify the same code to get to achieve the required task.

Interface Programming: Coding

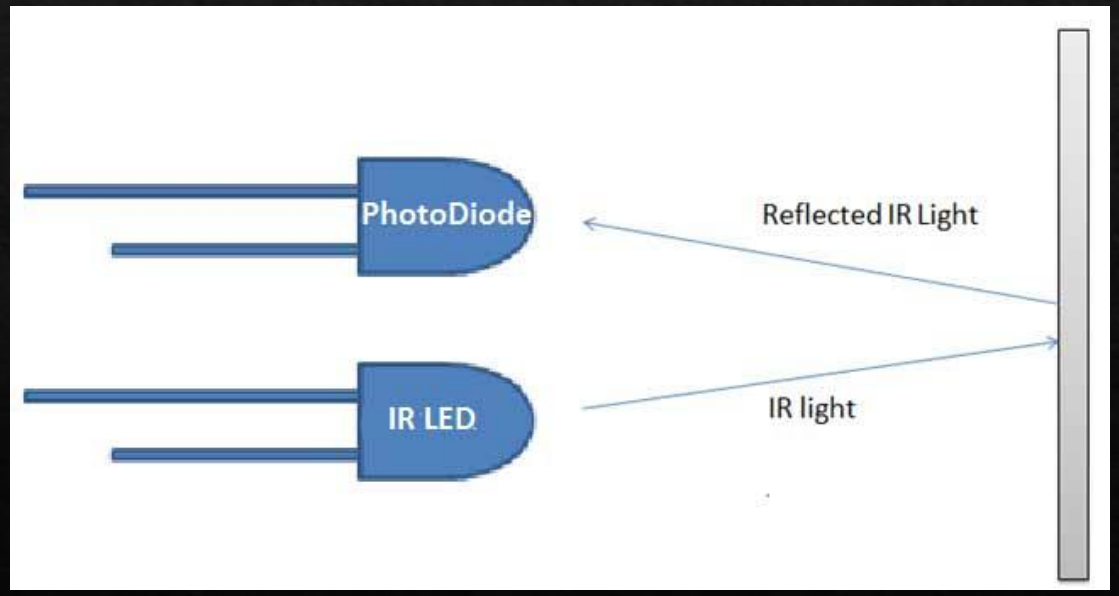
```
 GNU nano 3.2                    IR_LED/IR_LED.c

#include <stdio.h>       // Standard Input Output Operation
#include <unistd.h>     // POSIX API library for suspending operation
#include <iobb.h>       // Accessing BB-WI GPIOs

// Program begins
int main()
{
   int counter=0;      // Counter Initialise

   // Initialise the GPIO Library
   iolib_init();

  // Set the direction of pins
   iolib_setdir(9,15,DigitalIn);
   iolib_setdir(8,11,DigitalOut);
   pin_low(8,11);         //Set pin P8.11 low

   // Continuous forever loop
   while(1)
   {
      // Check if sensor high
      if(is_low(9,15))
      {
         pin_high(8,11);
         //for(i=0;i<=15;i++){    }
         counter++;
         printf("Letter # %d\n",counter);
         usleep(120000);      // Suspend process for 0.12 seconds
      }
      if(is_high(9,15))
      {
        // If no obstacle do nothing
      }
   }
iolib_free();    // Free the GPIOs once done returns 0 on success -1 on fail
return(0);
}
                        [ Read 37 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Wire Interface: IR Sensor

- For the connection of the hardware we need to look first look at the working of the sensors understand the working of their pins. Until now we just know what are the pins avaliable and the voltages required or avaliable across the pins.
- The IR (Infrared) Sensor basic application is either for level detection or for proximity detection. It is more famous used in line follower robots.
- These sensors consists of an IR LED and IR Photodiode with the sensor circuit build either with the help of an op-amp or timer IC 555. In this project we have used the sensor which already been built on PCB for using the timer IC 555. The timer IC in this case has been configured in Astable Mode (a.k.a Frequency Generator)

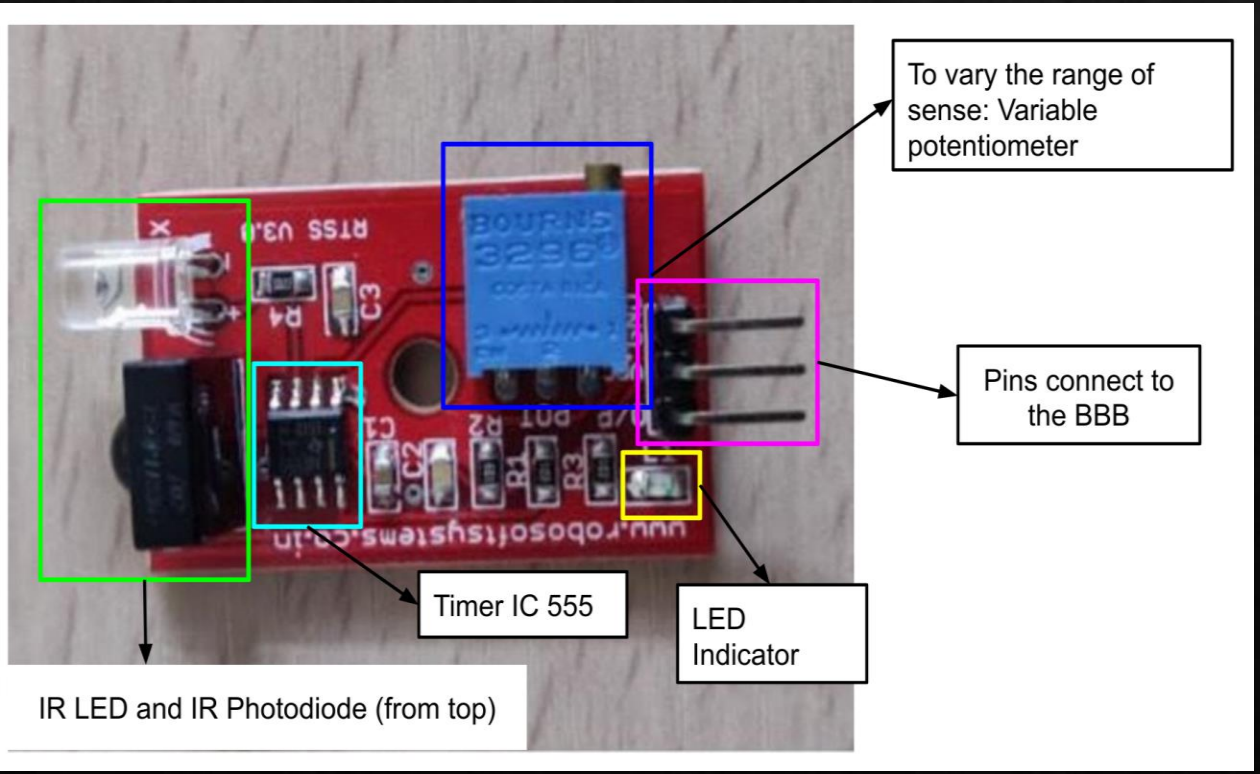
Wire Interface: IR Sensor

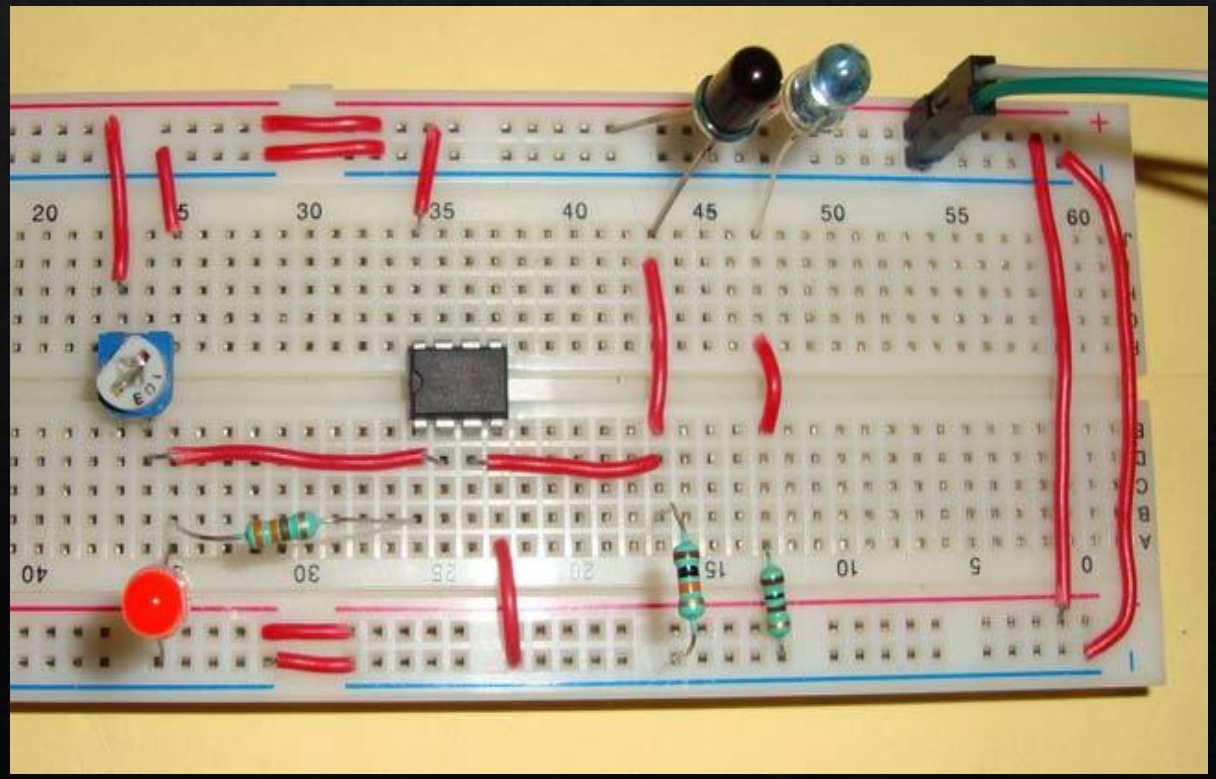
- IR sensor is governed by following three laws namely: Planck's Radiation Law, Veins Displacement Law and Stephen-Boltzmann Law
- The basic concept of an Infrared Sensor which is used as Obstacle detector is to transmit an infrared signal, this infrared signal bounces from the surface of an object and the signal is received at the infrared receiver.
- The sensor we have an active sensor behaving active high i.e. when an object is not present it will appear high



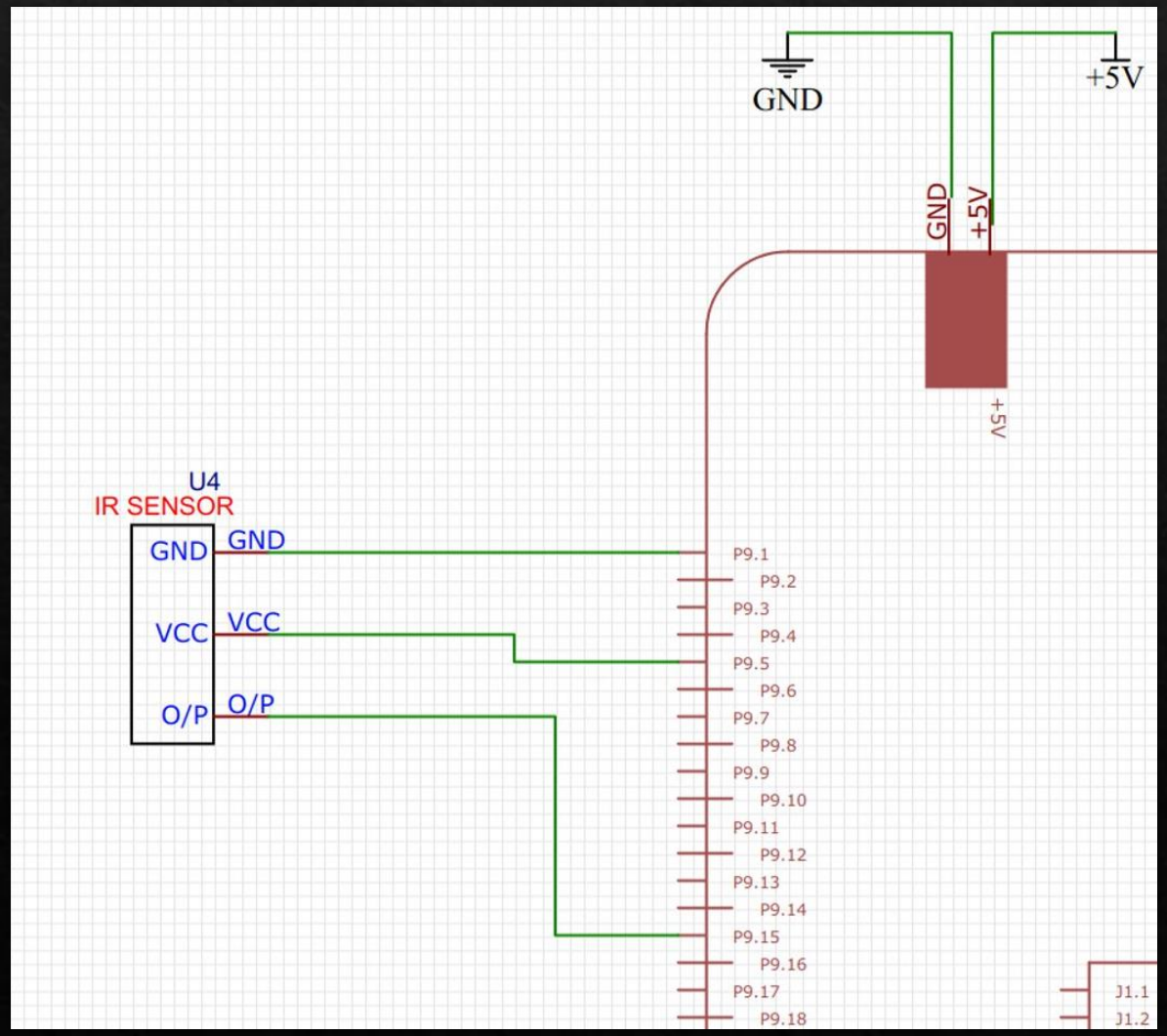
Wire Interface: IR Sensor

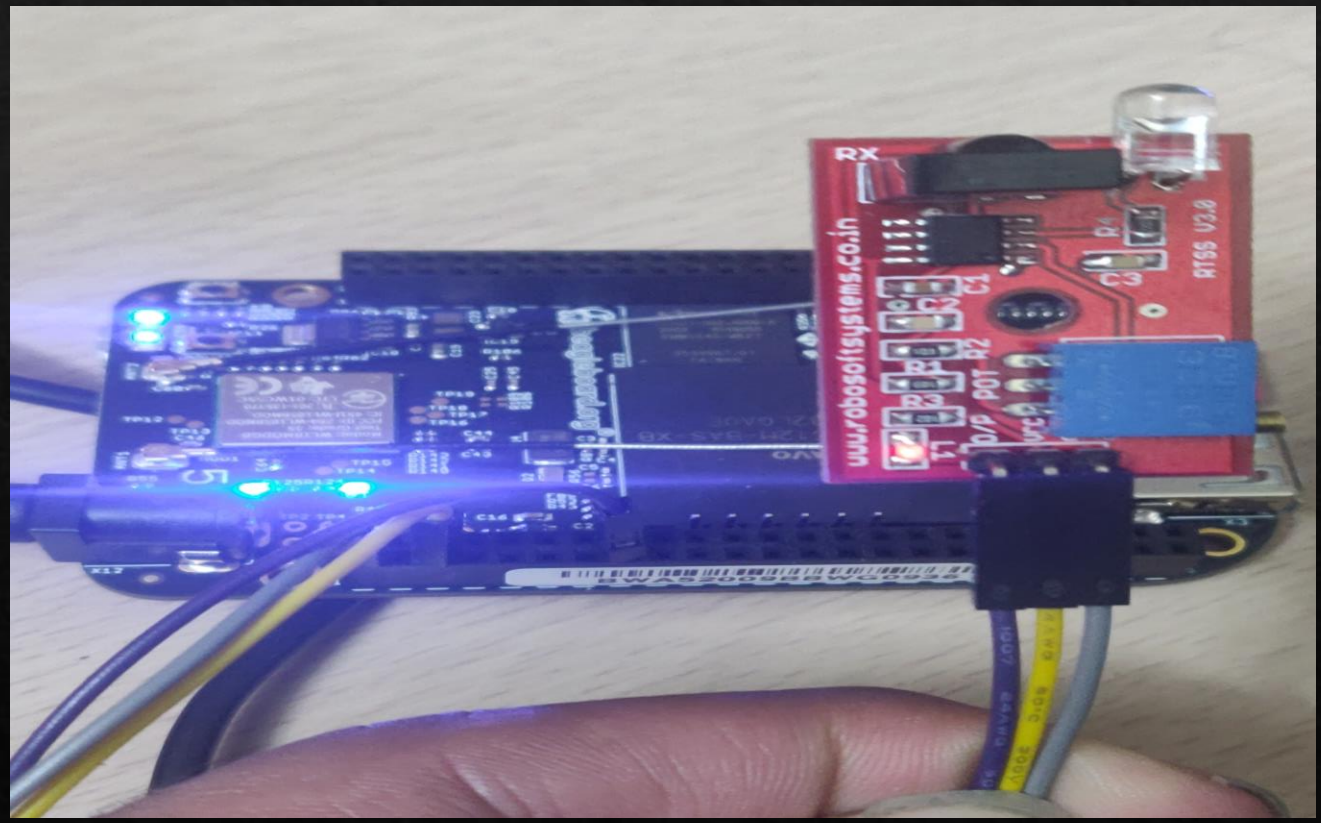
- We now look at the sensor module and how the raw connections can be made.

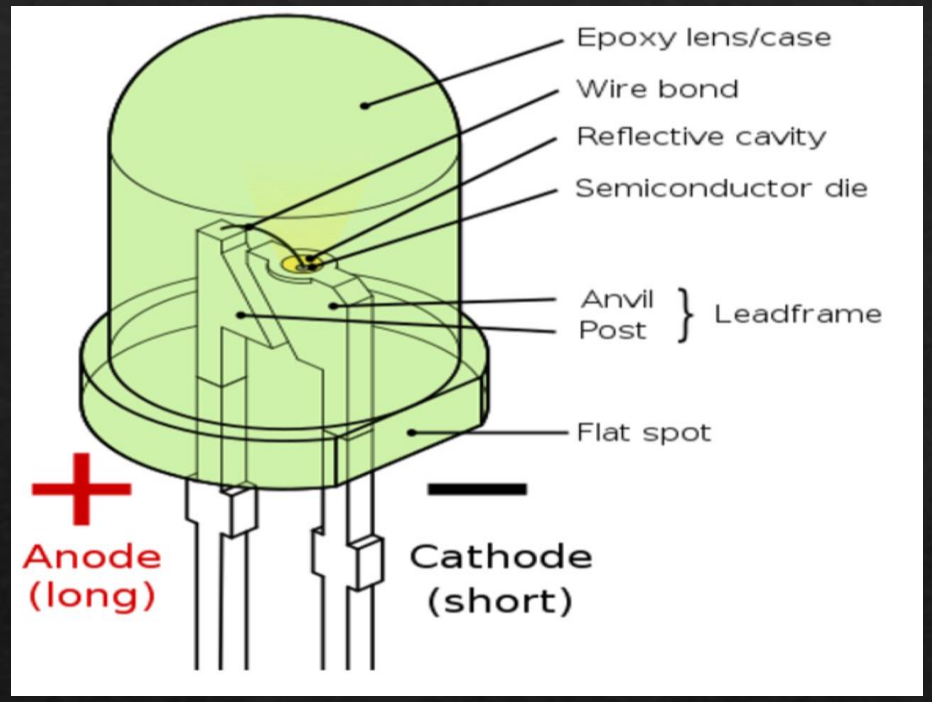


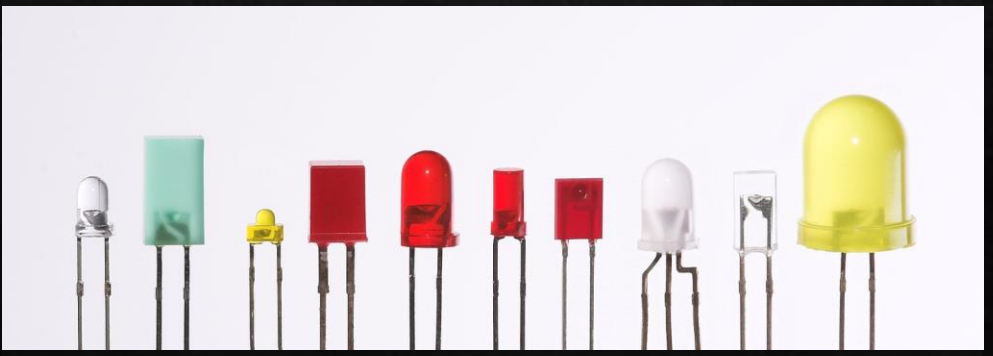


Wire Interface: IR Sensor and BB-WI





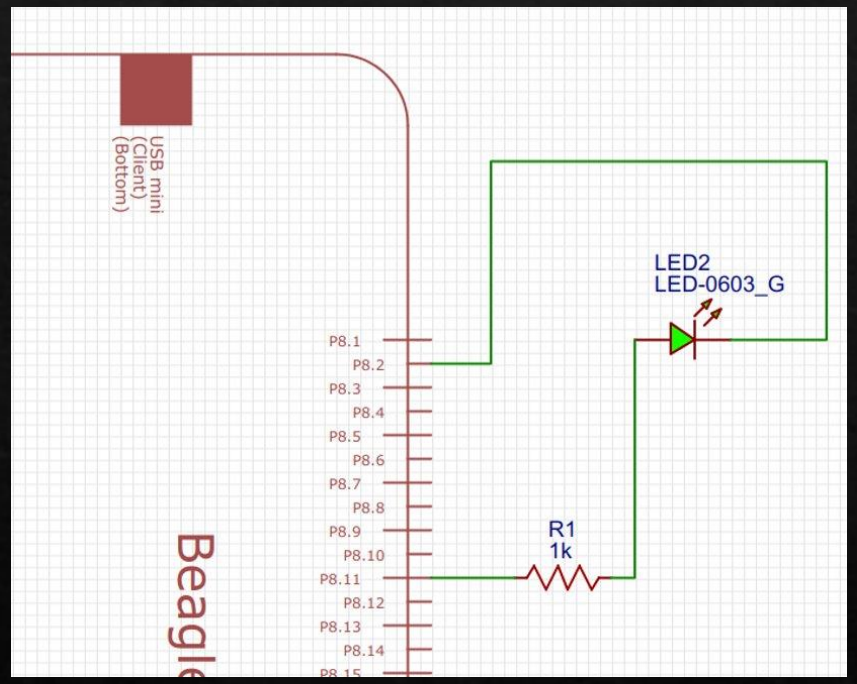


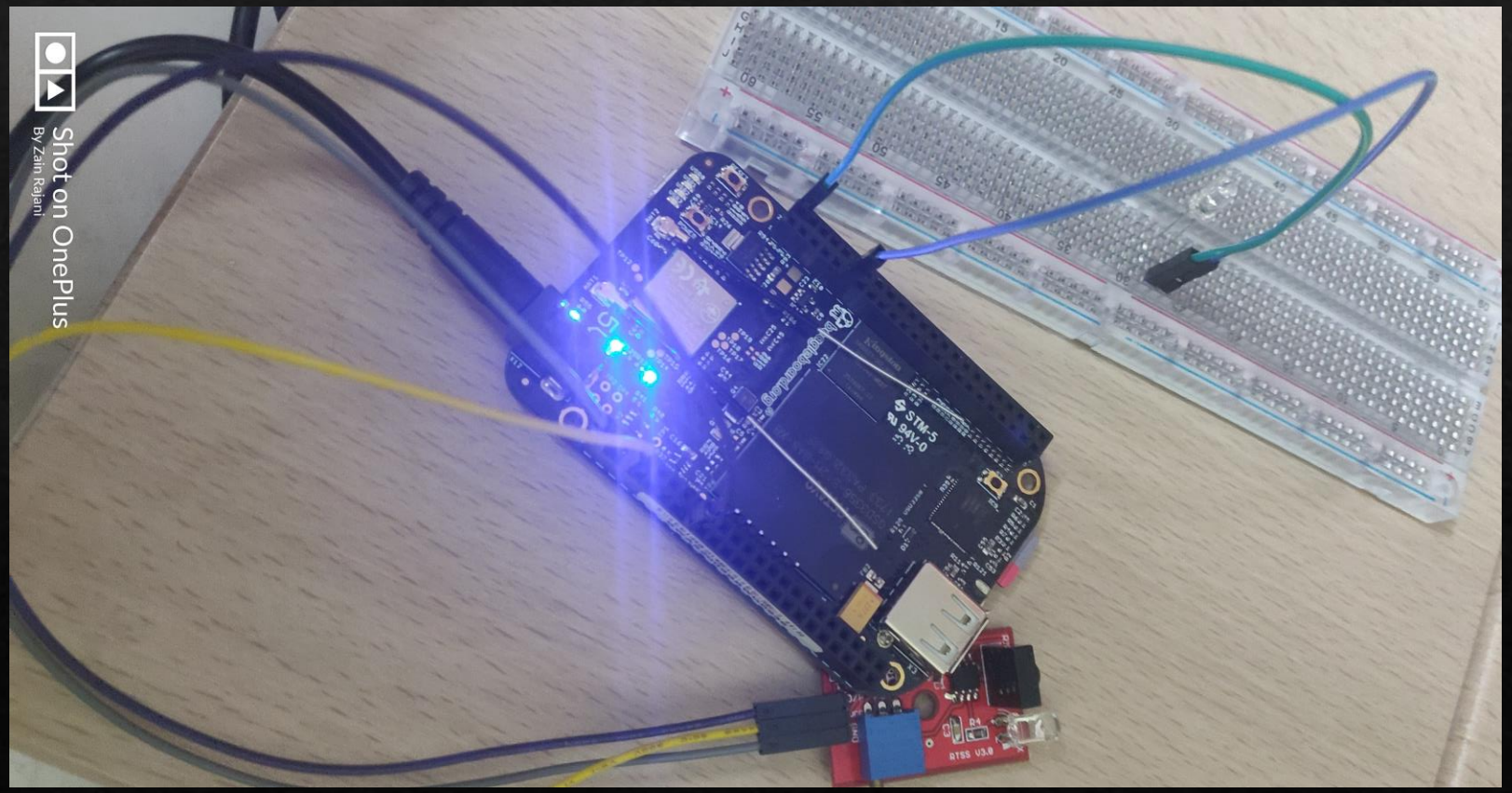


Wire Interface: LED

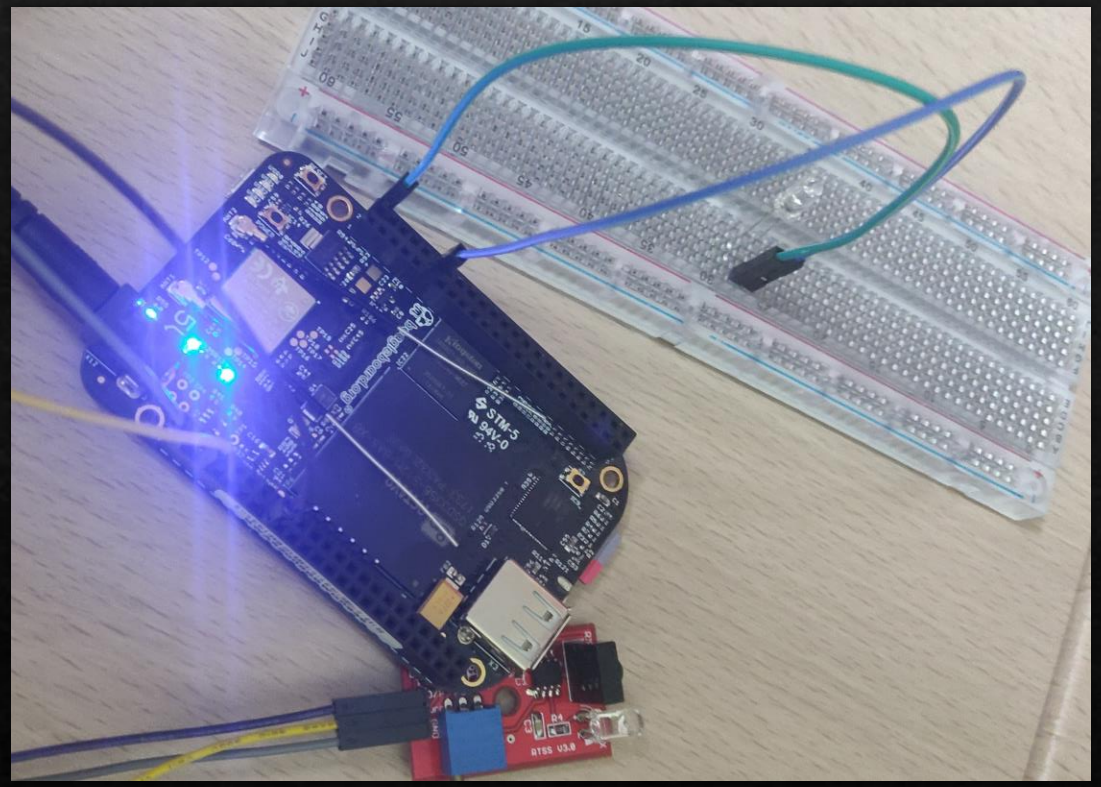
- A light emitting diode (LED) is a semiconductor light source that eits light when current flows through it. Electrons in the semiconductor recombine with the holes releasing energy in the form of photons.
- The colour of light is determined by the energy required for the electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light emitting phosphor on the semiconductor device.

Wire Interface: LED, IR Sensor with BB-WI

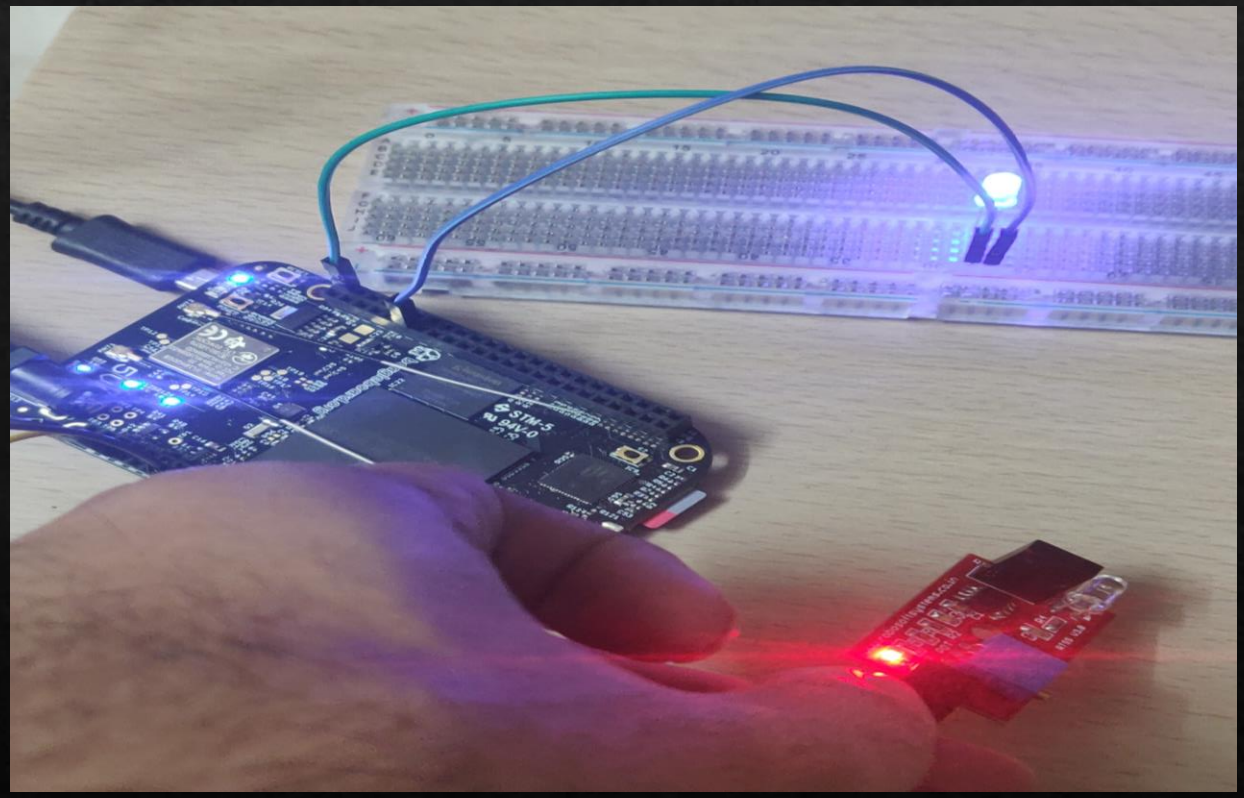




Execution

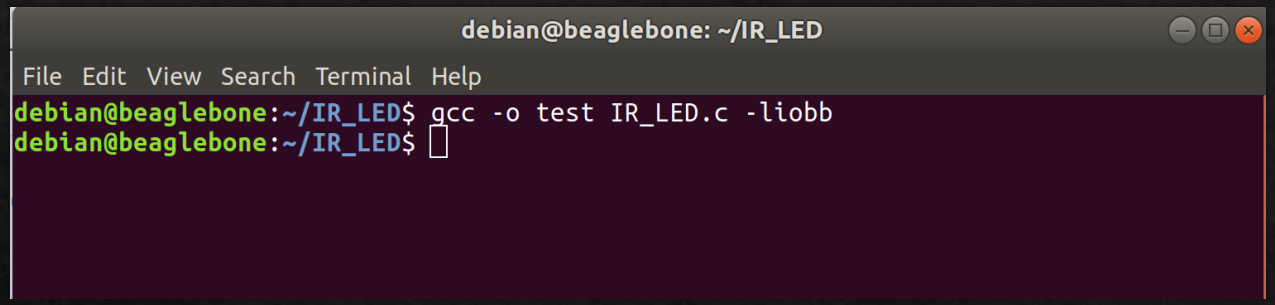


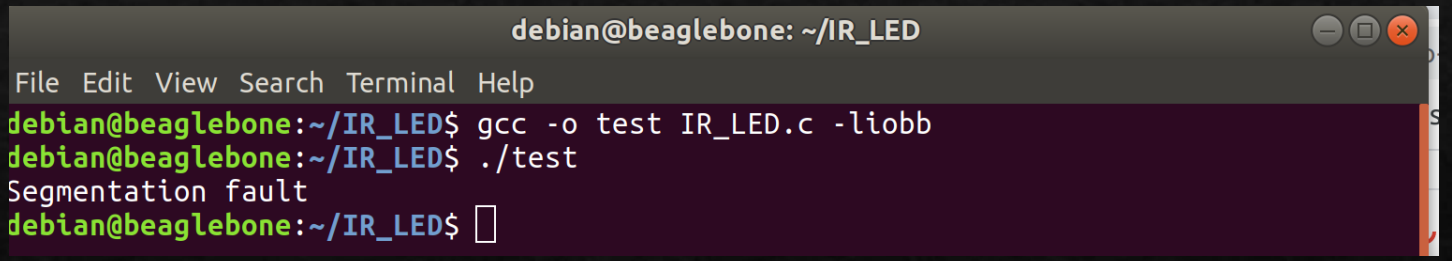
Initial State when the program begins to run

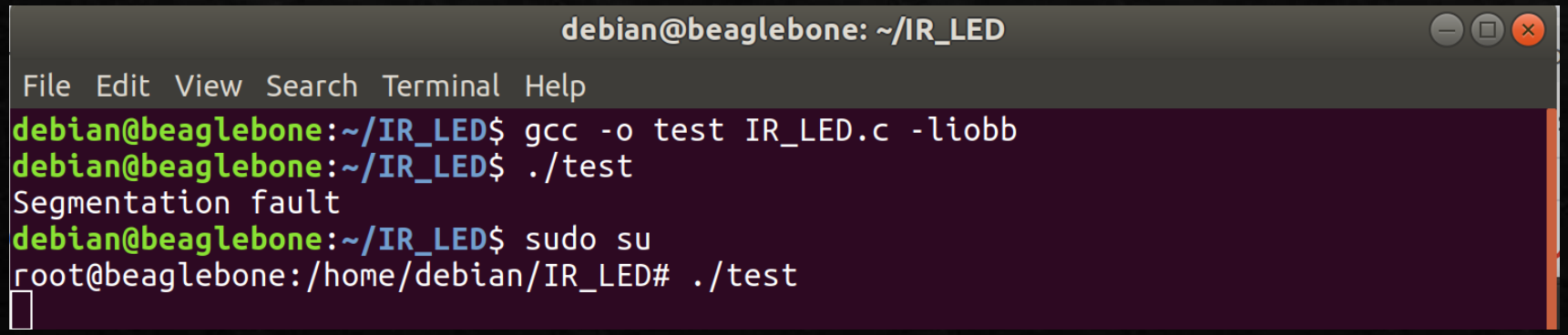


When the letter arrives in the letter box

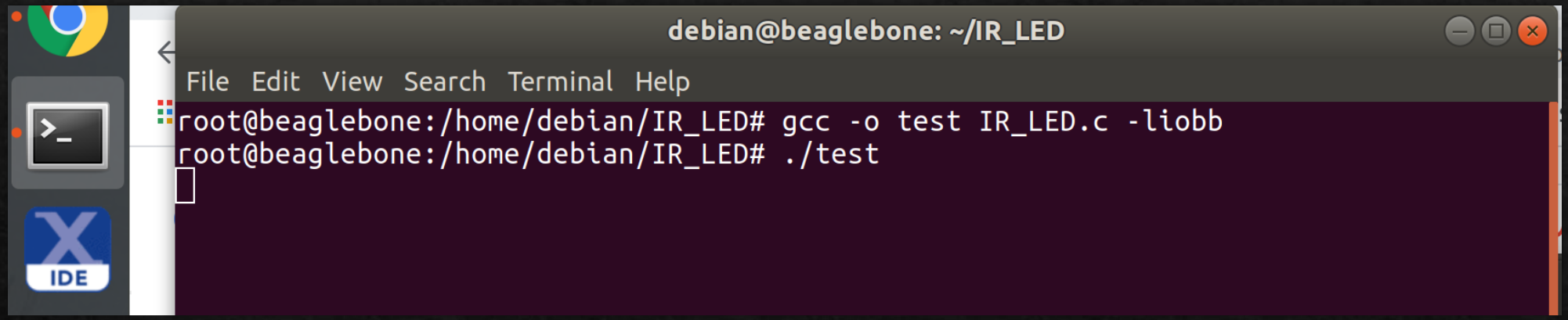
Execution

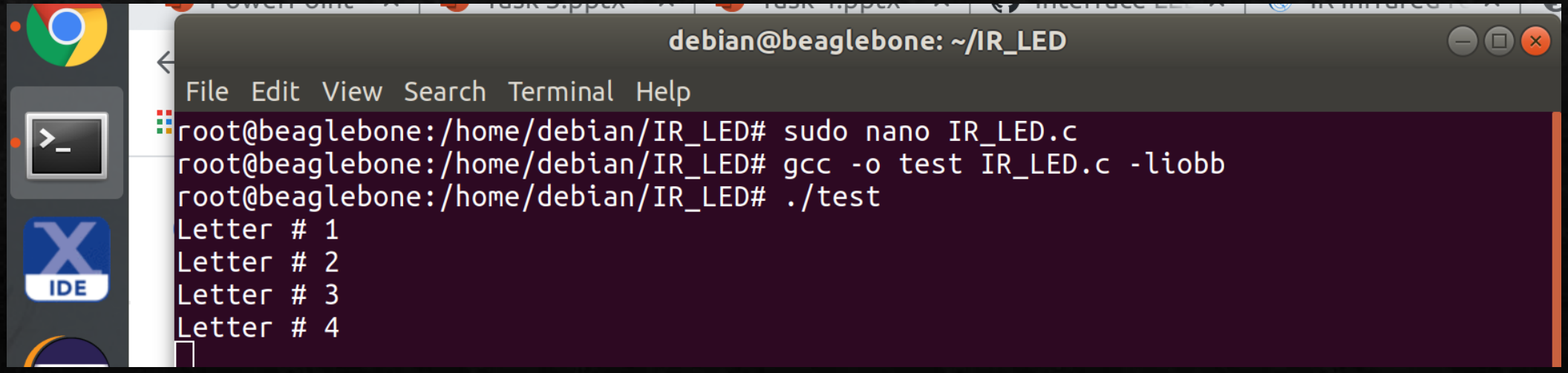

debian@beaglebone: ~/IR_LED
File Edit View Search Terminal Help
debian@beaglebone:~/IR_LED$ gcc -o test IR_LED.c -liobb
debian@beaglebone:~/IR_LED$


debian@beaglebone: ~/IR_LED
File Edit View Search Terminal Help
debian@beaglebone:~/IR_LED$ gcc -o test IR_LED.c -liobb
debian@beaglebone:~/IR_LED$ ./test
Segmentation fault
debian@beaglebone:~/IR_LED$


debian@beaglebone: ~/IR_LED
File Edit View Search Terminal Help
debian@beaglebone:~/IR_LED$ gcc -o test IR_LED.c -liobb
debian@beaglebone:~/IR_LED$ ./test
Segmentation fault
debian@beaglebone:~/IR_LED$ sudo su
root@beaglebone:/home/debian/IR_LED# ./test

Execution


debian@beaglebone: ~/IR_LED
File Edit View Search Terminal Help
root@beaglebone:/home/debian/IR_LED# gcc -o test IR_LED.c -liobb
root@beaglebone:/home/debian/IR_LED# ./test
IDE


debian@beaglebone: ~/IR_LED
File Edit View Search Terminal Help
root@beaglebone:/home/debian/IR_LED# sudo nano IR_LED.c
root@beaglebone:/home/debian/IR_LED# gcc -o test IR_LED.c -liobb
root@beaglebone:/home/debian/IR_LED# ./test
Letter # 1
Letter # 2
Letter # 3
Letter # 4
IDE

Troubleshooting

- During the interfacing or implementing the project one may encounter many issues and one must try to resolve it if possible. Thus, the same occurred with us as well. We had some issues facing with it and we list some of them here and how did we overcome them.
- For the hardware connectivity we were good to go as the connection was already been tested with the arduino in our previous meetings thus connection was not a problem for us.
- But for the software since now we had to connect everything with the Beaglebone Wireless we faced some issues like running the code, getting the required libraries and controlling the sensor through programming.

Troubleshooting

- At the time when one had to download the required GPIO libraries on the Beaglebone itself as we were using nano and GCC complier thus we wanted the libraries to be present on the BB-WI and for the same reason we required the internet connectivity.
- Thus, we made several attempts to connect it but later realised that since the router was far away from the BB-WI we had to go a little close in distance and hence it connected perfectly.
- When one wishes to run the code he/she shall must be logged in as a root user (super user)
- The reason being since we were trying to get the access some files which require adminstrative permission i.e. we were trying to change or set the functionality of the header pins which can be done only from the system files and thus we required full permission for it and this can be obtained only by being the admin/ root user.

Troubleshooting

- At the final stage since the IR sensor had to count the number of letters thus, we had to make sure that at a time only one is counted and not more than one at a time.
- But during the first run though the sensor was working properly counting but it used to count till the time the rays were interrupted. Thus to resolve we implemented delay by using the for loop.
- But later when thought of code optimisation which was one of the things mentioned in the proposal thus, we used the POSIX API usleep() function which can act like a delay and halt the system until the time mentioned in the bracket.
- The parameter of the usleep() is in microseconds. Thus, this helped us in suspending the current process and hence we were able to slow down the count

Conclusion

- At the end of the interfacing we were able to achieve the objectives that we set for ourselves at the start of this task.
- We were able to connect the sensor and LED appropriately as we had set in the schematic capture meetings
- The sensor counting the letter with a good amount of accuracy and the LED went HIGH when the first letter arrived in the box.
- Though the task were achieved successfully and as no project can be present without flaws thus, we found some flaws. For example, if two letters were deleivered it might be counted as one only.
- Also, since we use just document type of letters what would happen if the letter is of a big size or a box is delivered.

References

IR (Infrared) Obstacle Detection Sensor Circuit. (2017, December 24). Retrieved November 01, 2020, from https://www.electronicshub.org/ir-sensor/

Kumar, A. (2019, January 31). IR-Based Counter System with IoT. Retrieved November 01, 2020, from https://www.hackster.io/kashwani893/ir-based-counter-system-with-iot-6c69d0

Light-emitting diodes Circuit, Working Principle and Application. (2018, May 16). Retrieved November 02, 2020, from https://www.elprocus.com/light-emitting-diode-led-working-application/

Molloy, D. (2019). *Exploring BeagleBone: Tools and techniques for building with embedded Linux*. Indiapolis, Indiana: Wiley.

S. (2020, February 13). BeagleBone Black (BBB) and PocketBeagle I/O (GPIO), SPI and I2C Library for C – 2019 Edition. Retrieved November 02, 2020, from https://www.element14.com/community/community/designcenter/single-board-computers/next-genbeaglebone/blog/2019/08/15/beaglebone-black-bbb-io-gpio-spi-and-i2c-library-for-c-2019-edition

Usleep(3) — Linux manual page. (2017, September 15). Retrieved November 01, 2020, from https://man7.org/linux/man-pages/man3/usleep.3.html